Collaborative Policy Administration in Online Social Networks

Supriya V. Pawar ME(Student)
Department of Computer Engineering
Sinhgad Academy of Engineering, Pune University.
Pune, India
dubalsupriya@gmail.com

Asst. Prof. L. J. Sankpal
Department of Computer Engineering
Sinhgad Academy of Engineering, Pune University
Pune, India
ljs.sae@sinhgad.edu

Abstract— Collaborative Policy Administration is one of the efficient methods of policy administration in order to protect sensitive data loss. In Collaborative Policy Administration, a policy administrator can refer to other similar policies to set up their own policies to protect privacy and other sensitive information. In this work an Improved Collaborative Policy Administration is proposed and being evaluated for more effective application of Collaborative Policy Administration to ensure improved security in many networking applications. In Improved Collaborative Policy Administration, in order to obtain similar policies more effectively, a text mining based similarity measure method is utilized. The major enhancements proposed in Improved Collaborative Policy Administration over Collaborative Policy Administration are the investigation of safety definitions using real time environments. Additionally efforts are being made to improve permission model method of Collaborative Policy Administration to achieve effective policy administration.

Keywords*- Improved Collaborative Policy Administration, Refine, Verification, Policy Enforcement Point, Policy Decision Point*

*****

I. INTRODUCTION

All every software application domain should provide and make sure security issues. For the most part, as countries around the world transition from paper-based to electronic information record infrastructures, compliance with these data protection laws will require sophisticated information management technologies. Technical and policy challenges in relation to the widespread adoption of electronic information records systems have been discussed. There are also different aspects between the users and the service providers. The majority of users want to disclosure only least privacy data, and the service providers request at most personal information. Under this situation, if most right of information management comes up to the service providers, it provides the unfair position to the users. It is the drawback of monopolistic information management technologies. Security issues usually are derived from laws such as data protection acts or general security rules branching from the domain itself. However, the laws and rules are given as plain texts and lack a common formalism. This may make it impossible to predict unambiguous privacy regulation and privacy guidelines.

During the last few years ontology has been used as formalism to describe laws and rules on common bases. Ontologies have proven to be a useful tool in the areas of the personalized information management and semantic web. Using ontologies, user provide a common description for any type of policy, rule, and law, independently from the specifics of the system implementation.[2]

The policy-based management is extensively used technique to deal with complex and large-scale network systems.[1] Traditionally, construction of policy-based management consists of four core components as in Fig 1: policy enforcement point (PE), policy decision point (PD), policy repository (PR), policy administration point (PA) as shown in figure. The policies in PA are specified and verified by policy administrator or group and also the policies in PR are deployed by them. Once the system runs, the applicable policies from PR will be retrieved by PD and conclusion will be made. In case the subject wants to open a file (authorization action) or launch a logger to record system context (obligation action), PE takes control of the decisions.

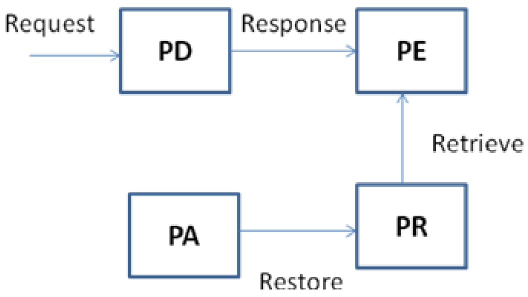

Fig 1: Policy Driven Management Architecture

In Collaborative Privacy Management System (Co- PMS) each user provides own privacy policy by using a policy creation interface, as in Fig.2 This interface is useful in making the specific privacy policy ontology for the each user. Each individual stores the encrypted privacy policy in his/her potable storages or in his/her computer. And then they can use it as new authentication certification. The policy negotiation engine performs collaborative process with the certain service's data disclosure policy. For reflecting current legislation law of ontologies is referred. When the privacy policy makes agreement with the data disclosure policy, then policy negotiation engine sends the encrypted results to application services. The result is used as a consensual privacy policy and also can used a kind of certification for legal user.[4]

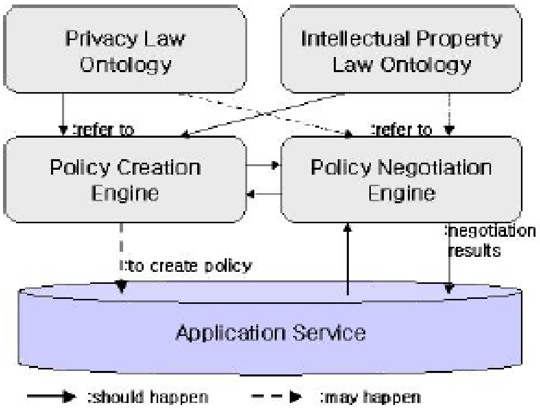

Fig. 2 Policy Based Privacy Management System

Architecture

In case of application of the policy-based management method to emerging applications such as mobile and social network services, if more privileges than those are normally essential by a subject are assigned by not well-trained administrator, cause real troubles. In Android application development, three responsibilities are usually involved in the policy administration: Application Developers declare the permissions requested by the application; Application Marketers verify if the application is lawful; Application Users decide about approval of the consent. These three roles are typically performed by those who are not well trained in policy-based management i.e. the developers usually proclaim more permissions than necessary because of their inclination to make the development of applications easier or even misinterpretation of technical documents. The marketers generally have a tendency to permit more applications despite of the wicked permission requests; and additionally, the application users approve all the requests due to their excitement to use the application, without realizing what the requested permission means. The social network services also face the same issue, wherein a user is asked to grant access to private data to third-party applications. This challenge to policy administration is ever more severe due to the outburst of these applications.

To address the challenges in policy administration, Improved Collaborative Policy Administration (ICPA) method has been projected and explained in this study. The basic principle of ICPA is that applications with similar functionalities shall have similar policies. Accordingly, ICPA will scrutinize the policies already specified by similar applications and execute collaborative recommendation, to specify or verify policies. Predefined algorithms such as a category-based algorithm or a text mining-based algorithm can be utilized to calculate extent of similarity.

II. RELATED WORK

A Policy as follows: "A policy is a set of rules reflecting an overall strategy or objective, affecting the behavior of agents and thus designed to help control and administer a system".[5]

Initially an agent-based architecture, which encompasses several agents that work together to provide the Policy Management services to the applications. In this architecture, and the Policy Service Agent and the Policy Management Agent are two agents of particular interest. The role of Policy Management Agent is defining, editing, storing and assigning policies. To get done this task, the Policy Management Agent may access the application profile accumulated in the Policy Information Base. The responsibility of the Policy Service Agent is to carry out the task of interpreting and enforcing policies. This requires a continuous communication between the application and the policy service agent. This communication is utilized to negotiate and exchange policy information updates that would impact the behavior of the agents representing the application at the run-time.

Traditional policy system has following demerits:

i. The marketers on average tend to permit more applications regardless of the malicious permission requests. The application users may not know what the demanded permissions mean, thus approving all requests because they are eager to use the application. The same issue exists in social network services, where a user is asked to grant access to private data to third-party applications. This challenge to policy administration is increasing severe due to the blast of these applications.

ii. In accumulation to this CPA method is having some shortcomings which need to be address in near future such as safety definition is not properly investigated and evaluated; permission model presented in CPA needs to improve.

A. Co-PMS Architecture

The entire Co-PMS is comprised of three stages policy creation stage, policy negotiation stage and lastly application service retrieval as in Fig.3.

- *Policy Creation Stage*

In the policy creation stage Fig.3, there are two components of the policy - the privacy policy and data disclosure policy. Service providers create the data disclosure policy through the policy creation engine. The policy manages the access privileges for each role according to the category of information feature, the purpose of request, and the projected recipient of results.

Using the policy creation engine user creates the his/her own privacy policy. This encrypted policy controls the leakage of his/her personal information, also according to the category of information feature, the intended recipient, and the purpose of the request.

The policy creation engine has three main elements such as a policy creation interface, an ontology interpreter, and a policy creator. With the help of policy creation interface users were easily define their privacy policy without the expertise. The ontology interpreter imports the privacy law ontology and the intellectual property law ontology. After integration of the rules from ontologies policy creator provides encrypted privacy policy.

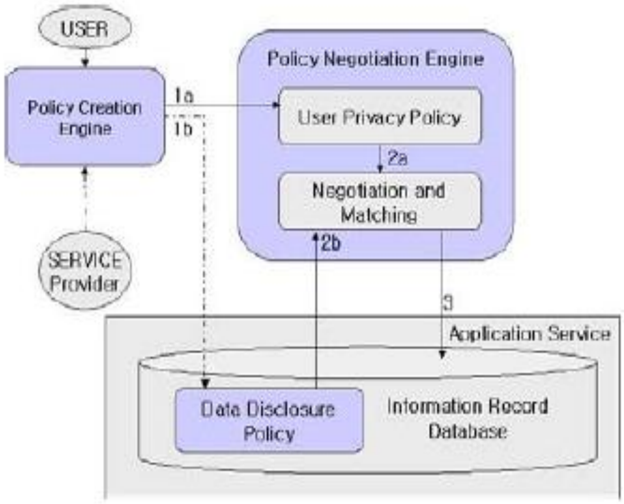

Fig 3: Collaborative Privacy Management System Architecture

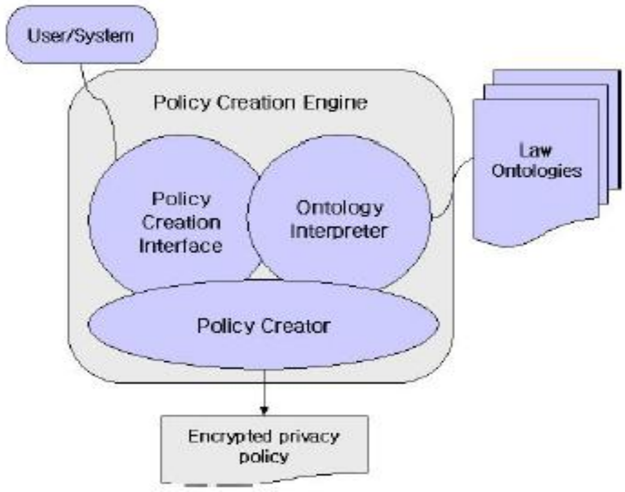

Fig 4: Policy Creation Engine

- *Policy Negotiation Stage*

In the policy negotiation stage In Fig.3, the user is informed to the system organization‘s policies concerning data use and disclosure, advised of any disagreement with one’s own privacy and security preferences. This fully automated process is completed before the user provides any personal data to the organization. The user first uses the policy creation engine to convey his personal policy concerning the use and disclosure of his personal data. This information is matched with the system organization’s privacy and security policies to identify any conflicts. The user gives the suggestion of these conflicts and given a chance to resolve them or terminate the process. At the end , the user should update the policy regarding whether his data may be disclosed to third parties or utilized for a different purpose than for which it was accumulated. This modified information are recorded as a result of policy negotiation in the application service database. It is factored in at the time of service processing. The policy negotiation engine has mainly four elements such as a policy reader, a policy analyzer, negotiation processor, and result creator, as in Fig.5.

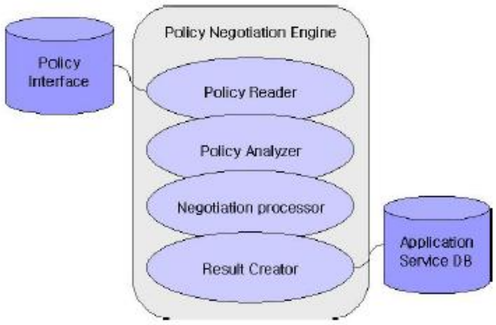

Fig 5: Policy negotiation engine

The reader imports the privacy policy and the data disclosure policy. The analyzer matches for each entry of the policies. If there are some disagreements, the negotiation processor provides a disagreement-report and sends it to user and service provider. When the processor get a reply from user and service provider, then the result creator makes an agreement result. This result could be a policy agreement between the policies.

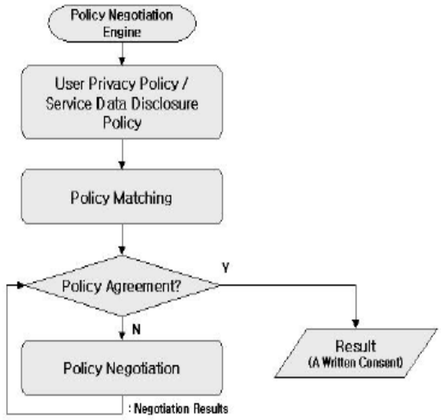
Policy Negotiation Engine
User Privacy Policy / Service Data Disclosure Policy
Policy Matching
Policy Agreement?
Y
N
Policy Negotiation
Result (A Written Consent)
: Negotiation Results

Fig 6: Negotiation flow

Before the user provides any personal data to the organization, the negotiation engine performs this fully automated process. A successful policy negotiation confirms agreement between the data disclosure policy and the privacy policy concerning the processing of the personal data. This agreement is important for personal certification to access a certain service.

- *Application Service Stage*

In the application service retrieval stage in Fig.3, the application system controls accesses based upon the user’s purpose, role, and intended recipient. The service system apply the result of negotiation as a kind of user’s certification , user’s information, and a agreement between service provider and user. This system already installed the result of policy negotiation between the privacy policy and the data disclosure policy. Through a database interface this Co-PMS can be integrated into existing environments.

III. PROPOSED SYSTEM

In this study, the Improved Collaborative Policy Administration (ICPA) is presented. As the vital idea of CPA is that applications with related functionalities shall have similar policies, ICPA will examine policies already specified by other similar applications and perform collaborative recommendation. The extent of similarity will be calculated by a text mining based algorithm.

The major enhancement proposed in ICPA over CPA is the investigation of safety definitions using real time environments, such as online social networking datasets. Additionally, efforts are being made to improve permission model method of CPA using fine grain access control. Also more security will be provided for History Policy Base (HPB) by the administrator.

IV. BACKGROUNG AND MOTIVATION

A. No trusted Administration Point

A professional expert or group will take charge of the policy administration in the traditional administration model.
However, upcoming applications, especially social network services mobile applications, face up to the existing trust model in the policy administration. A developer of a third party application must request the privileges to be used by the application and may not know what is at risk if an application requests privileges. To achieve more usage of any application, normally the end user will grant all requests from third-party applications

The second aspect of the changing trust model, there is conflict of interest in the involved party members. The developer to run his or her application with less privilege restrictions, so a developer is tend towards request more privileges.

The third aspect of the changing trust model is that the vigor of different supervisors is totally different. Usually a supervisor could verify the all policy requests.
Thus result of the changing trust model, overclaim of privileges is widespread in these upcoming applications. This breaks the basic security principle that is principle of least privilege. Thus, the changing trust model is the novel strategy to strengthen the policy administration.

B. Motivated Scenario:

- **Android Application:**

In the Android security framework, a developer sets the permissions for an application requested by various users. End user decides whether the requested permissions are legal for his or her mobile device. Due to the openness of the Android security framework, hundreds of millions of developers and users are involved. Though there is no tools' support, the developers could misunderstand the description of the requested permissions. As a result, the overclaim of permissions is widespread in Android applications. On the other side, the end users normally grant all requested permissions due to the lack of knowledge of policy administration. So, many applications with malicious behaviors can be uploaded to markets and spreads across all over the Android devices.

ICPA will provide two functions to help marketers, end users and application developers of Android, which is collaborative policy design and collaborative policy verification. Collaborative policy design support the application developers in deciding permissions to achieve the applications' functionalities and observing the principle of least privilege, along with ensuring the normal functionalities of the applications. Secondly, collaborative policy verification search the end users with malicious permission requests, so privacy leakage or financial losses are occur.

- **Social Network Services:**

In social network services, third-party web-based applications could request susceptible information of end users. The end user approves sensitive requests one by one is allowed in social network services. The developer can decide which sensitive requests can be set according to other similar policies. So, he or she can develop securer and more satisfactory applications for end users.

V. IMPROVED COLLABORATIVE POLICY ADMINISTRATION

A. ICPA Model:

The proposed Improved Collaborative Policy Administration consists of two main stages, collaborative policy design and collaborative policy verification.

Definition 1: Collaborative Policy Administration Model is,

$$ICPA :\{ Admins, CDM, CVM\}$$

Here, Admins refers to all involved policy administrators, like end users, developers, and marketers in the Applications.
Definition 2: Collaborative Policy Design Model is,

$$CDM :\{ PBhist, SimF, SUB, RefF, \Delta, P_{ref}\}$$

A policy administrator Admins can acquire a refined policy set ⊆ Pref according to a refinement function ⊆ RefF, which is a refinement driven by history data.
In Definition 2, PBhist means to a policy base that contains a various policies previously created by administrator itself.
PB_{hist} refers as,

$$PB_{hist} := 2^{SUB\_PER}$$

Here, SUB means to the subjects in a system. For example, all applications belong to SUB. PER means to all available permissions. Also SimF selects similar subjects, then produce their policies according to the subject's attributes as the similar policies.
Formally,

$$SimF: SUB \times PB_{hist} \rightarrow P_{similar}$$

Here, $P_{similar}$ means to all various policies of the similar subjects. RefF means to the refinement functions, each one of which will output a policy set according to the attributes of a subject ∈ SUB, its similar policies ∈ $P_{similar}$, and δ∈Δ, which may be a number.

$$RefF : SUB \times P_{similar} \times \Delta \rightarrow P_{ref}$$

Here, P_{ref} is ⊆ SUB ×PER.

Definition 3: Collaborative policy verification model:

CVM :{PBhist, SimF, SUB, VeriF, VeriR}

A policy administrator ∈ Admins can obtain a verification result ∈ VeriR for a target policy set ∈ P_{target}, which contains all polices assigned to a target subject ∈ SUB, according to a verification function ∈ VeriF.

Here, SUB means to the target subjects that will be verified. VeriF means to the verification functions, each one of which will verify the target policy set, move towards a verification result.

VeriF:

$$VeriF: SUB \times P_{similar} \rightarrow VeriR$$

B. Enforcement framework

In Fig 7 a policy administrator can leverage the framework to administrate policies via a web browser phone or development tool. The direction for key data flows is nothing but direction of arrows. Similarity measure methods and the history policy base are two key components in the enforcement framework. To impose CPA, the administrator should arrange a sufficient number of policies at first. Collaborative policy design, collaborative policy verification are the two key functions provided by the framework. These two functions depend on the history policy base as well as similarity measure methods. Then obtaining the similar policies, the two functions call a refinement algorithm, a verification algorithm. Finally, collaborative policy design and collaborative policy verification will display the output to the administrator on various user interfaces like development tool, a phone, web browser.

C. Key Algorithms

To impose ICPA, similar policies algorithms, refinement algorithm, and verification algorithm are proposed as follows:

- *Similar Policy Algorithm:*

Each similar policies algorithm obtains a similar policy set according to an input subject. If for every policy in the HB, every similar policies algorithm decides whether its subject is similar to the required subject, then add it to the similar policy set.

A novel text mining technique to obtain similar policy sets of applications in Algorithm 1. This novel technique leverages the explanation of a target application to search similar applications, and then adds the requested permissions of the similar applications to the similar policy set of the target application. A TF-IDF method is engaged to create key words of application description, and then scores will be produced according to the key words. Finally, the novel technique chooses a predefined number (threshold) of applications according to the scores. At the end adds the chosen application policy configurations to the similar policy set.

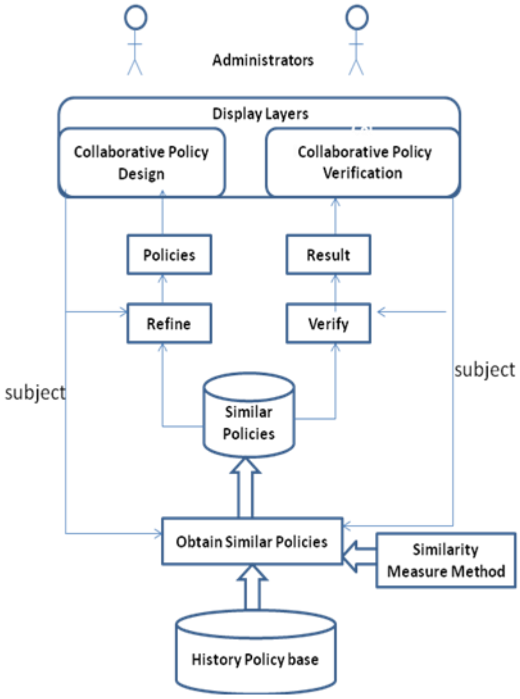

Fig 7 : Enforcement framework of ICPA.

***Algorithm 1**. Obtain Policies Based on Text Mining Method.*

Input:

```
subject ∈SUB
HB ∈ $PB_{hist}$
```

Output:

```
simpolicies ∈ $P_{similar}$
initialize ( )
query ← parse (subject.description)
for all subject ∈ HB do
    doc← subject.description
    score ← $a \times b \times \sum_{term\ \in query} (c \times d \times e \times f)(doc,term)$
    if score > simSubjs [simcountThreshold].score then
        simSubjs.removeLast ()
        simSubjs.insertIndescendingOrderByScore (subject)
    end if
end for
for all subject ∈  simSubjs do
    simpolicies.add (subject.permissions)
end for
return simpolicies
```

In this algorithm, the initialize function engage declaring the simpolicies, assigning 0 to the score of each element in simpolicies and building the index for all application description from HB if the index files are not available. The parse function tokenizes the explanation of the subject and returns a query object that is ready for searching. The statements inside the for loop are created by a typical text mining procedure based on TF-IDF. This method iterates on all subjects in the HB. simcountThreshold means to the threshold of similar subjects. If the score is higher than the score of the simcountThresholdth item in the similar subjects (simSubjs) then subject will be added into simSubjs in descending order by score. At the end all policies of each subject in simSubjs will be added to simpolicies. The O (n) is the time complexity of Algorithm1, where n means to the number of subjects in HB.

- *Refinement Algorithm:*

Algorithm 2 gives refinement policies according to a parameter δ, where δ is a number The time complexity is O (n), where, n means the number of policies in similar policies.

Algorithm 2. *Collaborative Policy Refinement.*

```
Input:
        subject ∈ SUB
        simpolicies ∈ $P_{similar}$
        δ ∈ Δ, it is a number
Output:
        refpolicies ∈ $P_{ref}$
        for all policy ∈ simpolicies do
        count [policy. permission] ++
        end for
        for all permission δ ∈ PERM do
        if count [permission]/simpolicies.size > δ
        policy.subject ← subject
        policy.permission ← permission
        refpolicies.add (policy)
        end if
        end for
```

- *Verification Algorithm:*

Algorithm 3 gives a quantified measure between the target policies and similar policies. Time complexity of this algorithm is O (n), where n means to the size of similar policies, because the size of similar policies is normally larger than the size of target policies. Also the step to fetch target policies can be optimized by using an index of subjects in HB. The final result is a vector of percentages, that means much percentage the permission of target policy take up in the similar policies. To get simplified final result, design an aggregation algorithm to achieve a single number rather than a vector.

Algorithm 3. *Collaborative Policy Verification*

```
Input:
        subject ∈ SUB
        simpolicies ∈ $P_{similar}$

Output:
        verifies ∈ VeriR
        for all policy ∈ simpolicies do
        count [policy.permission] ++
        end for
        targetpolicies ← ∀p ∈ HB: p.subject =
        for all tpolicies ∈ targetpolicies do
        verires [tpolicy.permission] ←
        count [tpolicy.permission]/simpolicy
        end for
```

VI. EXPERIMENTAL WORK

In ICPA to simplify the policy administration can refer to other similar policies to set up their own policies to protect privacy and other sensitive information. o obtain similar policies more effectively, a text mining-algorithm will be used. Then for enhancing design of policies refinement algorithm will be used. At the end to confirm the result verification algorithm will be used. Finally, collaborative policy design and collaborative policy verification will display the output to the administrator on various user interfaces like development tool, a phone, web browser.

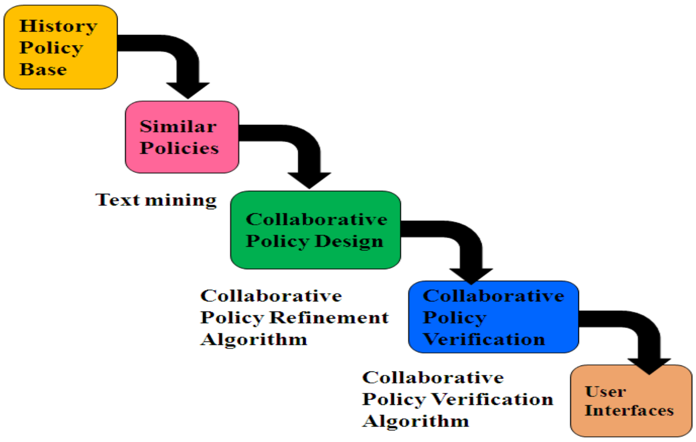
History Policy Base
Similar Policies
Text mining
Collaborative Policy Design
Collaborative Policy Refinement Algorithm
Collaborative Policy Verification
Collaborative Policy Verification Algorithm
User Interfaces

Fig 3 : Flowdiagram of ICPA.

CONCLUSION AND FUTURE SCOPE

The work presented in this paper proposes a novel policy administration mechanism, ICPA, to meet the requirements of the changing trust model, which has led to the widespread overclaim of privileges. ICPA leverages the similar policies to design or verify a target policy set, and simplifies the policy administration. This work provides definition of the formal model of ICPA and also the design of enforcement framework. Additionally, proposes text mining-based method of similarity measure to obtain similar policies.

For future scope Safety definition is investigated and evaluated to improve permission model. For analysis of ICPA, more strengthening is required for mathematics depth.

ACKNOWLEDGMENT

It is great pleasure for me to acknowledge the assistance and contribution of number of individuals who helped me in presenting “IMPROVED COLLABORATIVE POLICY ADMINISTRATION” I take this occasion to thank God, Almighty for blessing me with his grace and taking my endeavor to a successful I take this occasion to thank God, Almighty for blessing me with his grace and taking my endeavor to a successful culmination. I extend my sincere and heartfelt thanks to my esteemed guide, Prof. L. J. Sankpal for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected Head of the Department Prof.

B. B. Gite, for allowing me to use the facilities available. I am also thankful to other teaching and non-teaching staff of Computer Engineering Department and Library for their co-operation and help.

References

[1] IEEE transactions on parallel and distributed systems, 2013 Collaborative Policy Administration" by Weili Han, Member, IEEE, Zheran Fang, Laurence Tianruo Yang, Member, IEEE.

[2] "A Policy Management System for Collaborative Applications "by Mouhsine Lakhdissi, Hamid Harroud, AhmedKarmouch, Cliff Grossner, IEEE 2001.

[3] 2008 International Conference on Information Security and Assurance "Collaborative Privacy Management System" by In Joo Jang, Wenbo Shi, Hyeong Seon Yoo.

[4] International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Orlando, Florida, USA,

[5] A survey on policy languages in network and security by Weili Han Chang Leianagement.

[6] IEEE on knowledge and data engineering, vol. 25, no. 7, july 2013" Multiparty Access Control for Online Social Networks" by Hongxin Hu, Member, IEEE, Gail-Joon Ahn, Senior Member, IEEE, and Jan Jorgensen

[7] A.K. Bandara, N. Damianou, E.C. Lupu, M. Sloman, and N. Dulay, "Policy Based Management," Handbook of Network and System Administration, Elsevier , Nov. 2007.

[8] D. Verma, "Simplifying Network Administration Using Policy-Based Management," IEEE Network, vol. 16, no. 2, pp. 20-26, Mar./ Apr. 2002.

[9] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-Based Admission Control," RFC 2753, no. 2753, 2000.

[10] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "Policy Core Information Model—Version 1 Specification," IETF, RFC 3060, http://www.ietf.org/rfc/rfc3060, Feb. 2001.

[11] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android Security," IEEE Security & Privacy, vol. 7, no. 1, pp. 50-57, Jan./ Feb. 2009.

[12] B. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android Permissions: A Perspective Combining Risks and Benefits," Proc. 17th ACM Symp. Access Control Models and Technologies, pp. 13-22, 2012