**Collaborative Policy Administration**

**ABSTRACT:**

Policy-based management is a very effective method to protect sensitive information. However, the overclaim of privileges is widespread in emerging applications, including mobile applications and social network services, because the applications’ users involved in policy administration have little knowledge of policy-based management. The overclaim can be leveraged by malicious applications, then lead to serious privacy leakages and financial loss. To resolve this issue, this paper proposes a novel policy administration mechanism, referred to as collaborative policy administration (CPA for short), to simplify the policy administration. In CPA, a policy administrator can refer to other similar policies to set up their own policies to protect privacy and other sensitive information. This paper formally defines CPA and proposes its enforcement framework. Furthermore, to obtain similar policies more effectively, which is the key step of CPA, a text mining-based similarity measure method is presented. We evaluate CPA with the data of Android applications and demonstrate that the text mining-based similarity measure method is more effective in obtaining similar policies than the previous category-based method.

**EXISTING SYSTEM:**

The traditional framework of policy-based management consists of four core components policy decision point (PDP), policy enforcement point (PEP), policy administration point (PAP), and policy repository (PR). A well-trained policy administrator or group will specify, verify policies in PAP, and deploy the policies in PR. After a system runs, PDP will retrieve applicable policies from PR and make decisions. PEP takes charge of the decision, such as satisfying the request where a subject wants to open a file (authorization action), or launching a logger to record system context (obligation action). The overclaim of privileges, where a not well-trained administrator assigns more privileges than those which are normally required by a subject, is an increasingly serious problem, especially when the method of policy-based management is applied to emerging application scenarios, such as mobile applications and social network services.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Application users may not know what the requested permissions mean, thus approving all requests because they are eager to use the application.
* User will approve all requests from third-party applications, because User wants to run the applications, thus falling into the traps of malicious applications.
* The User leakage of their privacy.

**PROPOSED SYSTEM:**

This paper proposes collaborative policy administration (CPA). The essential idea of CPA is that applications with similar functionalities shall have similar policies that will be specified and deployed. Thus, to specify or verify policies, CPA will examine policies already specified by other similar applications and perform collaborative recommendation. The degree of similarity will be calculated by predefined algorithms, which could be a category-based algorithm, a text mining-based algorithm, novel method, enforcement framework and implement a prototype of CPA. The framework supports two types of user interfaces and provides functions of collaborative policy design and collaborative policy verification.

**ADVANTAGES OF PROPOSED SYSTEM:**

* Collaborative policy verification helps the end users identify malicious permission requests.
* Can develop securer and more acceptable applications for end users.

**SYSTEM ARCHITECTURE:**

**User**

🚺

**MODULES:**

* Collaborative policy design
* Collaborative policy verification
* Enforcement framework

**MODULES DESCRIPTION:**

**Collaborative policy design**

Here, *Admins* refers to all involved policy administrators, including, e.g., developers, marketers, and end users in the Android framework. policy administrator *Admins* can obtain a refined policy set *PSref* according to a refinement function. We design the policy using the system such as a new user can register and logins and upload any file. The user can design the policy in it. That is the policy may be like download option available or not, client details view options such that options.

**Collaborative policy verification**

A policy administrator Admins can obtain a verification result. VeriResultfor a target policy set PStarget, which contains all polices assigned to a target subject *SUBJS*, according to a verification function.

**ENFORCEMENT FRAMEWORK**

A policy administrator can leverage the framework to administrate policies via a phone, web browser, or development tool. The direction of arrows is the direction of key data flows. The history policy base and similarity measure methods are two key components in the enforcement framework. To enforce CPA, the administrator should prepare a sufficient number of policies at first. Furthermore, collaborative policy design and collaborative policy verification are the two key functions provided by the framework. These two functions depend on the history policy base and similarity measure methods. After obtaining the similar policies, the two functions call a refinement algorithm and a verification algorithm respectively. Finally, collaborative policy design and collaborative policy verification will display the results to the administrator on various user interfaces, e.g., a phone, web browser, or development tool.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.
* MOBILE : ANDROID

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/7.
* Coding Language : Java 1.7
* Tool Kit : Android 2.3 ABOVE
* IDE : Eclipse

**REFERENCE:**

Weili Han, Member, IEEE, Zheran Fang, Laurence Tianruo Yang, Member, IEEE, Gang Pan, Member, IEEE, and Zhaohui Wu, Senior Member, IEEE, “Collaborative Policy Administration”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014.