IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. IV (Jan – Feb. 2015), PP 18-23
www.iosrjournals.org

Corporate Policy Governance in Secure MD5 Data Changes and Multi Hand Administration

M.Saranya[1], S.Muthukumurasamy[2]
[1,2,] *(Dept. of Computer Science and Engineering, S.A Engineering College, India)*

Abstract: *Policy based management is an administrative approach that simplify the management of a given endeavor by establishing policies to deal with situation that are likely to occur. Most of the social network and mobile application in today's world define a very flexible policy that are used by user, easily which allows hacker to intrude in such social network and access user's private information, hence there is a need of strong policy for a social network application. The proposed approach verifies and analyzes the existing similar application and arrives at new policies by collaborating with the previous one to enforce security to the application and modification can be done with key generated by admin on permission by member. A text mining methodology proves to be simpler and stronger as more information about the application is not leaked out, it requires prior permission provided by user to track application information, thus policy admin forms an effective rule based system.*

Keywords: *Intruder, Policy based management, social networks, text mining.*

I. Introduction

Parallel computing is a collection of processing elements that communicate and cooperate to solve large problem fast. Distributed system is a collection of independent computers that appear to its users as a single coherent system. A parallel computer is implicitly a distributed system.

Collaborative Policy Administration (CPA) is one of the efficient methods of policy administration in order to protect sensitive data loss [1]. 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. Collaboration is a working practice whereby individuals work together to a common purpose to achieve business benefit. It enables individuals to work together to achieve a defined and common business purpose. It exists in two forms such as synchronous and asynchronous system. In synchronous system each user interacts in real time such as online meetings, through instant messaging, or via Skype. Asynchronous system each user interaction can be time-shifted, as when a user uploading the documents or annotations to shared workspaces, or making contributions to a wiki.

The Policy management is extensively used technique to deal with complex and large-scale network systems over traditional group based on policy management approaches which consists of four components [2]: policy decision point (PD), policy enforcement point (PE), policy administration point (PA), and policy repository (PR). a well-trained policy administrator or group will specify, verify policies in policy administration, and those policy are deployed in policy repository. After a system runs, policy decision will retrieve applicable policies from policy repository and make decisions based on the request. Policy enforcement 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).

II. Related Work

A Policy as follows: "A policy is a set of rules and principles, formulated and enforced by the governing body of an organization, to direct and limit its actions in pursuit of long-term goals."

Today computers are better understood and more economical, every day it brings new applications where those applications involve both storing information and simultaneous use by several individuals, for those applications system needs a desired authority structure [3]. These applications are agent-based architecture, where several agents that work together to provide the policy management services to the application. The policy management agent needs to be collaboration that is a group of people works together to obtain a common goal. The role of policy management agent is defining, editing, storing and assigning policies. 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.

For example, in credit bureau data bank system, is a company that collects information from various sources and provides consumer information with various activities where it perform various tasks like retail, business and commercial activities such as mobile bank, internet bank, self-service terminals, ATM, etc.., in

order to perform these activities, it uses policy management agent which defines policies in the following structure. Initially an applicant needs to fill the application form in order to become an authorized user, then these applications are validated and provide an identity to the user, these works are done by the data collection area once the data are collected, they are processed in the decision making area where the application are calculated, corrected based on the policy made by the credit bureau if the application is fraud they immediately reject the application. The policy rules used to refer manually to understand the applicant if the application is accepted then they is proceed to a final decision and conformation, once the application form is conformed then the contract is made with the applicant that is they can use the banking system for the business and commercial activity. The decision making model is processed as follows, initially following details like application details, internal account data and internal account score are integrated then process into logistic model, those data are given to decision score, then added with bureau data and bureau score in order to make decisions for the policy created, these policy management give rise to the following benefits such as increase predictive power of scoring model, Can manage huge data's of debtors and their performance.

Other examples of this system require protection of information are encountered every day: Airlines reservation system, law enforcement information systems; time-sharing service bureaus; on-line medical information systems; and government social service data processing systems. These examples span a wide range of needs for organizational and personal privacy. All of this system needs to have a central controlled sharing of information among multiple user, they require some plan in order to implement correct authority structure.

III. System Architecture

The major problem in the administration is to protect the policy from intruder hacking the data. In order to protect the data from the malicious modifications some policies should be used, such as authentication with password, secret question, or even fetching the MAC address from the system and also have communication protocol between the server and database clients.

Data is the most crucial part of an organization to protect the data from external attacks and also by the insiders. Standard database security mechanisms, such as access control, authentication, and encryption these are to protect data from external attacks when it comes to insiders, this paper provides a novel approach called multi-hand administration of a database system. The main idea is to provide a public key for any major data definition language (DDL) trigger technique. The administrator correlates both public and private key using database administrators enables the database repository change which prevents the malicious modifications from the intruder.

The keys are generated using multi-hand administration that is public key is generated by one of member and private keys are generated by the database administrated, these key are transferred by using .net SMTP and it is tracked in the database repository here the system is secured by hiding some necessary connection string information followed by the profile loading concept.

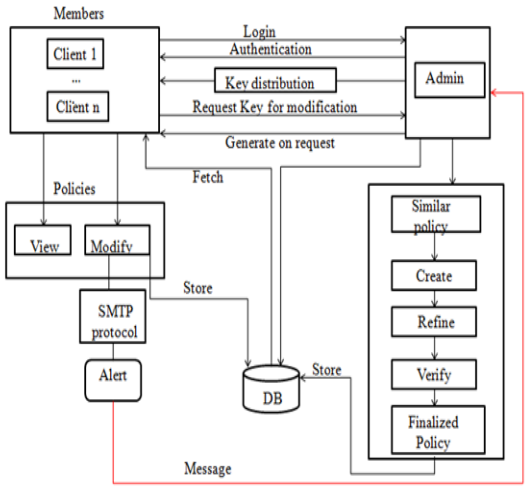

Fig 1: System Architecture.

This paper also uses SMTP (Simple mail transfer protocol) is a one of the application layer of the TCP/IP protocol. It simplifies the communication between email server that is done by using POP3 and IMAP protocol. It works closely to a Mail Transfer Agent (MTA), [6] which send your communication to the right email inbox using "store and forward" method, which is used to provide an alert message to administer in order an intimate that intruder is trying to modify the content in the data.

In fig 1 it illustrate a new users can be a member of the case with admin knowledge that is a member needs to login. Initially the user login to the system by providing their username, passwords and they set a personal question and answer that question so if they forget their password using this secret question they can generate a new password and in this login form it also fetches the MAC address is fetched from their system to provide security and it also helps in finding the intruders when some user tries to access the private data, then MAC address will be different, so they can't use those details, a onetime password is generated for each user while login in to the system a user is allowed to view the case details only after the approval by admin.

The admin creates policy by fetching similar policy using text mining algorithm, once a data is fetched they are modified according to the case details these cases are stored in the repository. These cases can be viewed by the member only after key generated by the admin using triple des algorithm, each user needs to provide their keys in order to view the data. Modification can be done only with the prior permission of the admin only when the admin allows accessing the data, user can modify it. If any user tries to modify the content, then an alert message is passed immediately to admin so that an intruder can be found easily this is done my simple mail transfer protocol.

IV. Policy Administration

In a policy-based management system, policy describes several actions and information according to the management. It defines an entity which is a set of related attributes. Executors and targets are subset of entities. Executors and targets are different roles that an entity can play in a policy, and these roles can change in different policies [4]. Policy-based management systems usually provide flexibility of constructing complex policies, which allow users to define multiple actions in one policy file. Therefore, an entity can be both executor and target in one policy file. In order to simplify the problem, we divide long policy into small segments. We assume that one segment describes one complete action containing one executor, one target, and its context (in the form of constraints). Therefore, a policy segment is represented as a tetrad (Executor, Target, Action, and Context). In our policy model, a policy segment is the smallest functional unit. In order to present these components in logic expressions, we formally define several key elements in our policy model below. Table 1 describes these components of a policy.

The policy based management system it is based on the collaborative administration which consists of two main stages: Collaborative policy creation and Collaborative policy verification [7].

Table 1: General Policy Model

Attribute	A piece of information describing certain aspect of an entity.
Entity	A collection of attributes describing a complete element in an information domain.
Executor	Executor is an entity that performs an action on another entity.
Target	Target is an entity that receives the action from another entity.
Action()=Executor × Target	A process of an entity affecting another entity or itself.
Context	A policy segment includes all constraints on actions and entities.
Segment= (Executor, Target, Action(),Context)	Smallest functional unit in policy.

4.1. Policy Creation

In policy creation it has two components of the policy they are privacy policy and data disclosure policy. Service providers create the data disclosure policy through the policy creation engine. The policy manages access privileges for each member according to the severity, the purpose of request, and the projected recipient of results [5]. The low severity actions are the conservation action which may send an alert detail but doesn't proactively avoid intrusion. Likewise, high severity responses are aggressive actions which are capable of avoiding an intrusion proactively by dropping the request, revoking or denying the required privileges and disconnecting the user.

Using the policy creation engine user creates their own privacy policy. This encrypted policy controls the leakage of his/her personal information, 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.

4.2. Policy Verification

In policy verification stage, the policy is verified by administrator where the admins can obtain a verification result for the target policy set. Policy set contains all polices assigned to a target subject according to verification function. The verification function will verify the target policy set, and provide a verification result; this result includes a simple conclusion of verification result.

In the policy negotiation, 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 policy negotiation engine has mainly four elements such as a policy reader, a policy analyzer, negotiation processor, and result creator. 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. 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

V. Algorithm

To impose this concept, these algorithm are used similar policy creation algorithm, refinement algorithm, verification algorithm and encryption algorithm are used

5.1. Similar Policy Creation Algorithm

Similar policies algorithm obtains a similar policy set according to an input subject. It fetches every policy from the history policy base, this algorithm decides whether its subject is similar to the required subject, then add it to the similar policy set.

The text mining technique is used to obtain similar policy sets of applications. 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. 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.

Algorithm Obtain similar policies Based on Text Mining Method

Input:

Subject ∈ SUBJS

HPB ∈ $PB_{history}$

output:

Similarpolicies ∈ $PS_{similar}$

initialize ()

query← parse(**subject**.description)

for all subject ∈ **HPB do**

doc←subject.description

score←$\Sigma_{term \in query}$(doc,term)

if score>simSub[simcountThreshold].score **then**

simSub.removeLast()

simSub.insertInDescendingOrderByScore

end if

end for

for all subject ∈ **simSub do**

Similarpolicies.add(subject,permissions)

end for

return similarpolicies

5.2. Refinement Algorithm

The refinement algorithm is used to refine the policies which are obtained from similar policies. 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 Collaborative policy refinement

Input:

subject ∈ SUB

similarpolicies ∈ $P_{similar}$
δ ∈ Δ, it is a number

Output:

refinepolicies ∈ P_{ref}
for all policy ∈ similarpolicies do
count[policy.permission]++
end for
for all permission δ ∈ PERMS
do
if count[permission]/simpolicies.size>δ
policy.subject←subject
policy.permission←permission
refpolicies.add(policy)
end if

5.3. Verification Algorithm

In policy verification algorithm, policies are verified from the refined data, the policy is verified according to the organization request. If the policy is not up to the need then the policy is negotiated. It 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 Collaborative Policy Verification

Input:

subject ∈ SUB
similarpolicies ∈ $P_{similar}$

Output:

Verifies ∈ VeriR
for all policy ∈ similarpolicies do
count[policy.permission]++
end for
targetpolicies ←∀p ∈ HB:p.subject=
for all policies ∈ targetpolicies do
verires[tpolicy.permission] ← count[tpolicy.permission]/similarpolicy
end for

5.4. Encryption Algorithm

5.4.1. DES (Data Encryption Standard) is a cipher block that uses shared secret encryption key. The DES is based on a symmetric-key algorithm that uses a 56-bits key. The encryption key is available for increasing computational power which made brute-force attacks as practical. The Triple Data Encryption Standard gives a comparatively simple method of increasing the key size of DES to protect against such problem, without the requirement to design a completely new block cipher algorithm.

5.4.2 Triple DES- Triple DES is also named as Triple Data Encryption Algorithm (TDEA or Triple DEA) cipher block, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. Triple DES uses a "key bundle" which comprises three DES keys, K1, K2 and K3, each of 56 bits (excluding unique or unity bits). The encryption algorithm is:

Ciphertext = EK3 (DK2 (EK1 (plain text)))

That is DES encrypts with K1, DES decrypt with K2, then DES encrypt with K3.

Decryption is the reverse:

Plaintext = DK1 (EK2 (DK3 (cipher text)))

It is decrypted with K3, encrypt with K2 then decrypt with K1.

Each triple encryption encrypts one block of 64 bits of data. In each case the middle operation is the reverse of initial and final. It improves the strength of the algorithm when using keying alternative 2, and provides backward compatibility with DES with keying alternative 3.

VI. Conclusion And Future Work

In CPA two approaches are introduced to improve the privacy policy management of data. First approach is Same as Policy Management, which leverages a user's memory and opinion of their friends to set policies for other similar policies. Text mining concept uses friend's policies and minimal task interruption to obtain substantial reduction in policy authoring times. In addition it perceived by user over traditional group based policy management approaches. Second approach is by fetching MAC Address from the user system and also providing one time password in order to secure the private details. Additionally, proposes to meet requirement of the changing trust model which leverages similar policies to design or verify a target policies set and simplifies the policy administration.

The policy set which is obtaining from collaborative policy design. This policy set, however, may be an intermediate result and can be adjusted to meet the requirements of system management or security management, the collaborative policy design should be accurate enough to be enforced. Thus, the refined policies can directly be used to determine whether the sharing operation should be allowed. An alert message can be passed when an intruder tries to modify the content using simple message transfer protocol (SMTP).

References

[1]. Weili Han, Zheran Fang, Laurence Tianruo Yang, Gang Pan, and Zhaohui Wu, "Collaborative Policy Administration" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014

[2]. Supriya V. Pawar, L. J. Sankpal," Collaborative Policy Administration in Online Social Networks "International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 2 Issue: 5

[3]. Jerome H. Saltzer, And Michael D.Schroeder," The Protection of Information in Computer Systems"

[4]. Vibha Verma, Mr. Avinash Dhole, "Analysis of Comparison Between Single Encryption(Advance Encryption Scheme(AES)) and Multicrypt Encryption Scheme

[5]. Gorrell P. Cheek, Mohamed Shehab "Policy-by-Example for Online Social Networks" SACMATO 12 JUNE 20-22, 2012 NEWARK, NEW JERSY,USA,ACM.

[6]. Kathy Wain Yee Au, Yi Fan Zhou, Zhen Huang and David Lie " PScout: Analyzing the Android Permission Specification " CCS'12.

[7]. William Enck , Peter Gilbert , Byung-Gon Chun , Landon P. Cox , Jaeyeon Jung , Patrick McDaniel , Anmol N. Sheth " TaintDroid: AnInformation-Flow Tracking System for Realtime Privacy Monitoring on Smartphones" To appear at the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10).