Secure Cloud Computing

Project Report

Arun Agarwal <axa103521@utdallas.edu>
Avinash Joshi <axj107420@utdallas.edu>
Shishir Krishnaprasad <sxk116430@utdallas.edu>

Introduction

This project is a Java based 'desktop application' that has admin page where you can add user, roles and map user to roles to resources. The basic functionality is to check that only qualifying user has access to resources based on the policy file. The engine also takes as input a Pig script (pre-defined) and executes it on data stored in HDFS, only if the querying user is allowed to access all input files specified in the script.

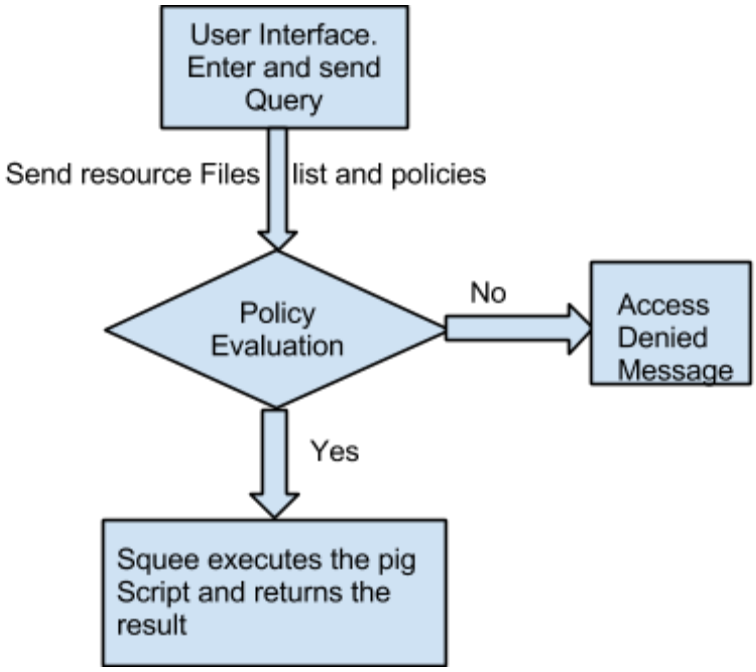
Problem Definition

In a distributed file system with multiple users, we need to ensure Role based fine grained access control on the resource of the file system. In our system we need a system to run PIG queries in hadoop environment with enforcement of Access Control on the resources present in the HDFS .

Our Solution

Flow Chart

Below is the architecture flowchart for the project.



Description

To enforce Access Control on the resource present in the HDFS, we used the Sun's implementation of XACML- eXtensible Access Control MarkUp Language, which uses the concept of Resource, Action and Role to enforce fine grained Role based access control on the various resource.

We generate a Role to user mapping for all the users in the system to provide different level of access control. In the UI we add new users to already existing roles or generate new Roles. These new Roles are merged with the existing file.

In our solution we do show the role to user mapping already assigned in the file.

For each of the resource file we generate a policyFile which specifies which Role have read access to a certain resource file, which gives us N policy files for N resources. When we change permission we need to take care that we re-assign permission to the all the users in the UI.

When a user Logs in to run PIG Queries, he gets the list of queries which are available for all the users, user selects one of the query and then we call the Sun's Policy Decision Point implementation of XACML where-in we pass the user credentials, role to user mapping ,the resource required by the user to execute the query and also the policy Files. The Policy Decision point then evaluates the request sent by the user and returns a boolean of true or false If the returned value is true then we call the PIG query execution engine to run the PIG query selected by the user else we return a error message of insufficient permission.

Components

User Interface

Login Screen

When application is run (details in README.txt), the user is prompted with a login screen (*fig. 1*).

Administrator UI

The default username password for AdminUI is root\toor. The AdminUI has three main functions as tabs:

> List Users:
> This lists all the users existing in the system (from etc/passwd file) and the list of roles with users in each role. A screenshot is shown in fig. 2.1.

Add user to role:

This tab allows administrator to add a new user to a role. Note that if a user already exists, it is added only added to the role. Refer fig. 2.2

Permissions:

Referring to fig. 2.3, this tab list all resources as a drop-down menu from the etc/resources.txt file. On the right is a table with the list of roles. The user can select a resource and the roles to which the resource file is accessible. Note that the role to resource mapping is recreated every time and not updated.

User UI

The user UI is show as screenshot in figures 3.1 and 3.2. Once the user is authenticated from the etc/passwd file, they are shown a screen with a dropdown menu with a list of functions that everyone can query. Upon clicking submit, the resource is checked against the resource to role mapping for all the resources in the pig query and if at least one resource is not permitted to be used by the logged-in user's role, a deny message (fig 3.2) is shown. Else the pig script is run and the output is shown in the display box (fig 3.1).

To run your custom PIG scripts, the user UI has the first dropdown as Custom

Pig Script Execution

Once the User UI determines that the current user has permission to access the resources, it calls PigExecute module and specifies the query file and mode (local/mapreduce). The PigExecute module then reads the queries from the file and adds them to a PigServer instance and executes them. The PigServer instance is told to store the output in the local HDFS. Once executed, it checks whether execution was successful. If yes, it reads the output from HDFS and returns it to the User UI module, where it is displayed in the text area.

Screenshots

Login UI

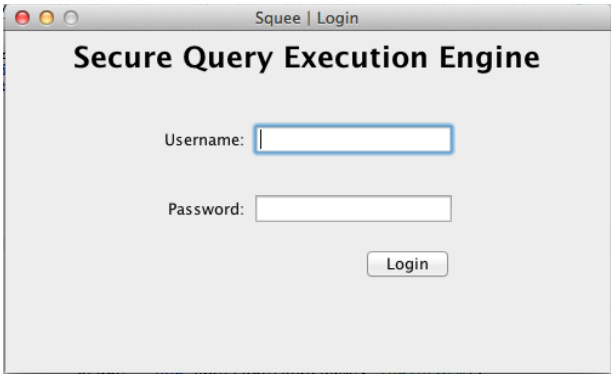


Figure 1: Login Screen

Administrator UI

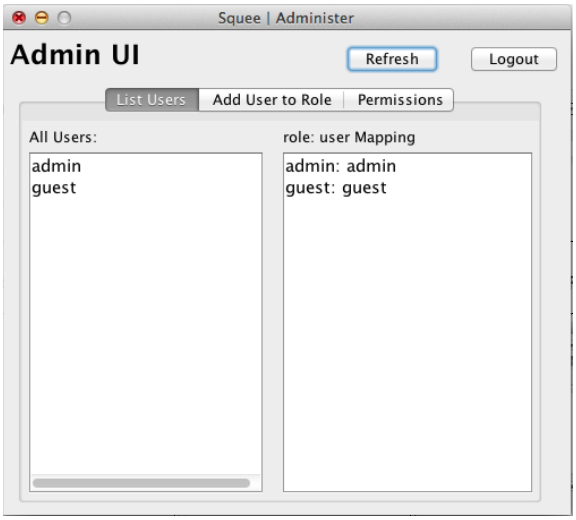


Figure 2.1: List of users and role

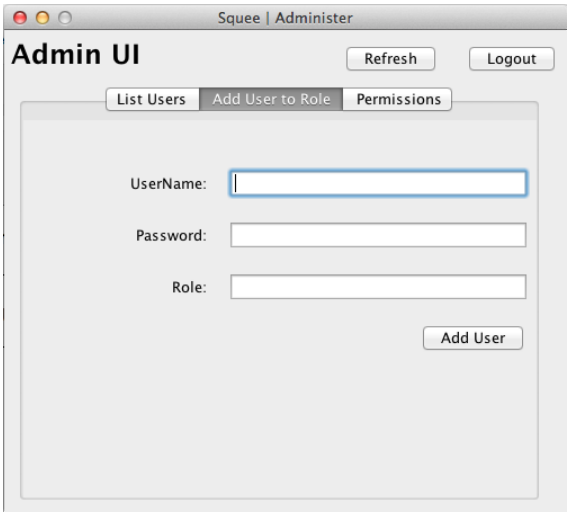


Figure 2.2: Create & add user to role

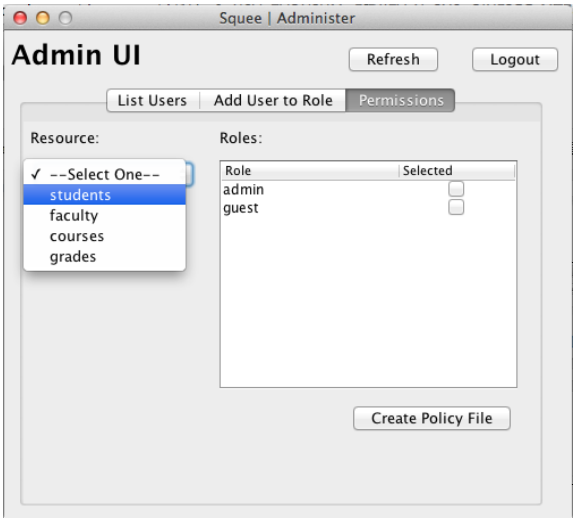


Figure 2.3: Create policy files

User UI

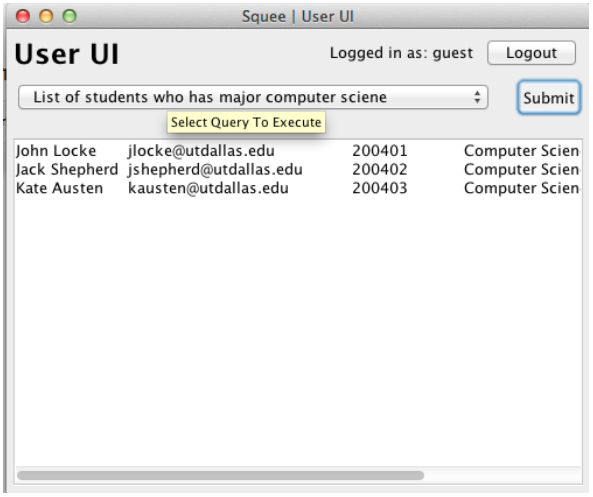


Figure 3.1: User with access to a resource

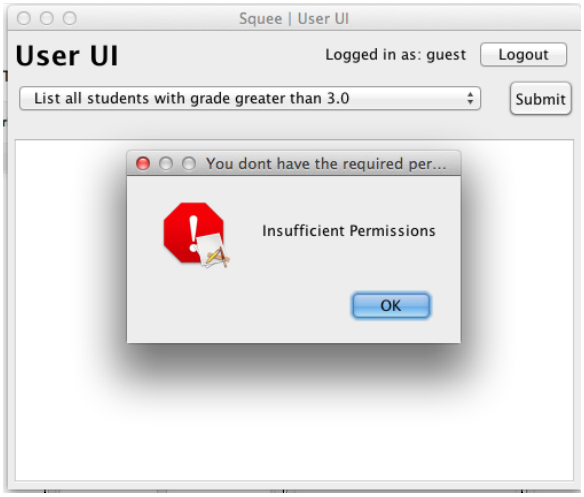


Figure 3.2: User denied access to a resource