BLOGGING WEBSITE

A CS814 Course Project Report

SUBMITTED BY

SANDEEP KUMAR MISHRA(202CS026)

ANKUSH RAVINDRA DHAMANWAR (202CS002)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, KARNATAKA

P.O. Srinivasnagar, Surathkal, Mangalore-575025

Karnataka, India

January 2021

Table of Contents

1.INTRODUCTION

Blogging has become such a mania that a new blog is being created every second of every minute of every hour of every day. A blog is your best bet for a voice among the online crowd. BIO&S "were usually tlæ work of a single individual occasionally of a small group, and often covered a single subject. More recently, 'multi•author blog;" (MARs) have developed, "with posts written by large numbers of authors and professionally edited. MABs from newspapers, other media outlets, universities, think tanks, advocacy groups and similar institutions account for an increasing quantity of blog traffic. The rise of Twitter and otlær 'microblogging" systems helps integrate MABs and single-author blog; into societal new streams. Blog can also be used as a verb, meaning to maintain or add content to a blog. A novel is a long, fictional narrative which describes intimate human experiences. WEBLOG is a combination of both Blog as well as Novels. Blog contains the Information of various things related to Technology, Education, News, Intemational, Business, Sports, Entertainment and ongoing college ætivities. The main aim of this project is to provide data to users in a single site. Users can gather the information from one site as well as give their feedback and create their own blog.

We have created a web app where any user can surf the page and read blogs posted by any user. We have a login system implemented in our app where users can register themselves on our site.

There are two types of users -

1. Admin: Admin is a unique and single user with all the rights and permissions of the web app. Admin can read all blogs, he can create blogs of his own. He can delete any blog of any user so that we can filter out useless or offensive blogs. He can also delete users.
2. Author: Author is a normal user with some rights and permission. He can read any blog of any user. He can create his own blog. He can also delete his own blog. Any person registered to this web app is assigned the author role by default.

TOOLS & TECHNOLOGY

1. HTML :

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within a tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human readable. Language uses tags to define what manipulation has to be done on the text.

2. CSS :

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

SQLite3:

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions and will store the data as-is if such a conversion is not possible.

Flask:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

AUTHORIZATION :

Role-based access control (RBAC) is a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments. A study by NIST has demonstrated that RBAC addresses many needs of commercial and government organizations. RBAC can be used to facilitate administration of security in large organizations with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication. Within an organization, roles are created for various job functions. The permissions to perform certain operations are assigned to specific roles. Members or staff (or other system users) are assigned particular roles, and through those role assignments acquire the permissions needed to perform particular system functions. Since users are not assigned permissions directly, but only acquire them through their role (or roles), management of individual user rights becomes a matter of simply assigning appropriate roles to the user's account , this simplifies common operations, such as adding a user, or changing a user's department. Role based access control interference is a relatively new issue in security applications, where multiple user accounts with dynamic access levels may lead to encryption key instability, allowing an outside user to exploit the weakness for unauthorized access. Key sharing applications within dynamic virtualized environments have shown some success in addressing this problem.

For a large number of users it is very difficult to make rules accordingly every user in such cases RBAC is very efficient since we are going to create some role and every user will assign to role according to their job.

Three primary rules are defined for **RBAC** :

1.Role assignment: A subject can exercise a permission only if the subject has selected or been assigned a role.

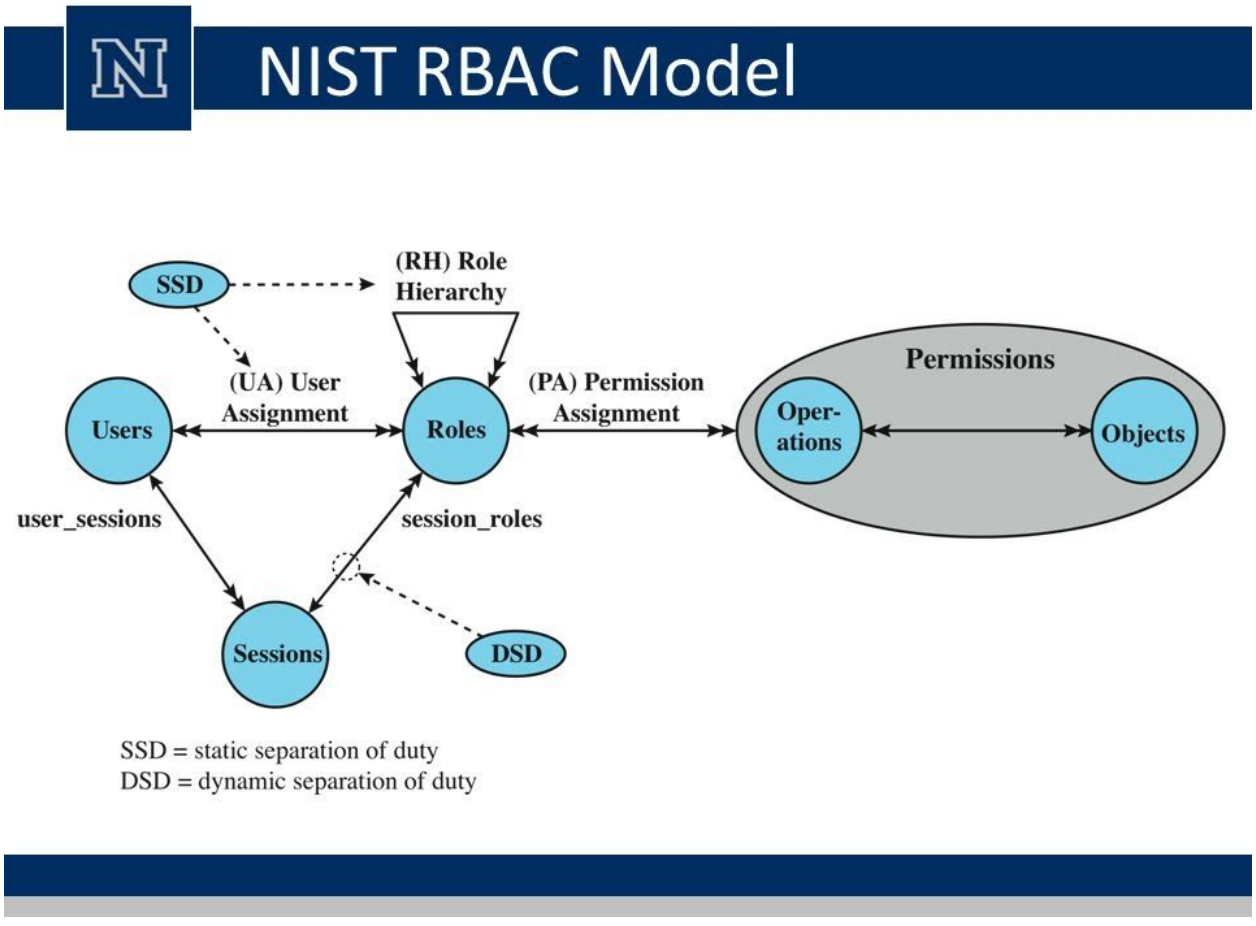
2.Role authorization: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.

3. Permission authorization: A subject can exercise a permission only if the permission is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can exercise only permissions for which they are authorized.

In our model we have two kind of roles:

1.**ADMIN ROLE:**Admin is a unique and single user with all the rights and permissions of the web app. Admin can read all blogs, he can create blogs of his own. He can delete any blog of any user so that we can filter out useless or offensive blogs. He can also delete users.

2. AUTHOR ROLE: Author is a normal user with some rights and permission. He can read any blog of any user. He can create his own blog. He can also delete his own blog. Any person registered to this web app is assigned the author role by default.



RBAC component Diagram.

RBAC POLICY

Additional constraints may be applied as well, and roles can be combined in a hierarchy where higher-level roles subsume permissions owned by sub-roles. With the concepts

of role hierarchy and constraints, one can control RBAC to create or simulate lattice-based access control (LBAC). Thus RBAC can be considered to be a superset of LBAC.

When defining an RBAC model, the following conventions are useful:

U = User or Subject = A person or automated agent

R = Role = Job function or title which defines an authority level

P = Permissions = An approval of a mode of access to a resource

SE = Session = A mapping involving S, R and/or P’

URA = User Role Assignment

PRA = Permission Role Assignment

RH = Partially ordered Role Hierarchy. RH can also be written: ≥ (The notation: x

≥ y means that x inherits the permissions of y.)

A constraint places a restrictive rule on the potential inheritance of permissions from opposing roles, thus it can be used to achieve appropriate separation of duties . For example, the same person should not be allowed to both create a login account and to authorize the account creation.

Thus, using set theory notation:

$PA \subseteq P \times R$ and is a many to many permission to role assignment relation.

$SA \subseteq S \times R$ and is a many to many subject to role assignment relation.

$RH \subseteq R \times R$.

CONCLUSION:

We have implemented the Role Based Access control (RBAC) using a simple web app of a Blogging website. We have created roles like admin and author with a set of policies. They are having certain permissions to access the functionalities of the web app.

We can still add more roles and assign different permissions to that role using RBAC. These roles can be handled and created by admin.

REFERENCES:

1. https://www.w3schools.com/
2. https://www.tutorialspoint.com/index.htm
3. S ANDHU , R. AND BHAMIDIPATI , V. 1997. The URA97 model for role-based administration of user-role assignment. In Database Security XI: Status and Prospect.
4. https://pythonhosted.org/Flask-Security/
5. https://getbootstrap.com/docs/5.0/getting-started/introduction/
6. https://www.tutorialspoint.com/sqlite/sqlite_python.htm