Aplicaciones Web

Autenticación de usuarios mediante ACL

5.2 SEGURIDAD: USUARIOS, PERFILES Y ROLES

Un aspecto complementario al manejo del estado en páginas web es el *control de usuarios*. Por lo general, las aplicaciones web están diseñadas para que usuarios con distintos niveles de acceso puedan realizar diversas operaciones, pero ¿cómo podemos organizar dichos niveles de acceso? Uno de los mecanismos más usuales es a través de *perfiles* o *roles*. Así, un usuario cualquiera que entre y utilice nuestra aplicación web. El rol es una de las características del usuario y es la propiedad que establece cuáles son las tareas o procesos que el usuario puede ejecutar. De esta manera, se crea un control de acceso jerárquico en el cual un usuario tiene un rol, un rol contiene varias tareas y una tarea es ejecutada a través de varios procesos.

5.2.1 LISTA DE CONTROL DE ACCESO (ACL)

Una vez visto el uso de sesiones, podemos pensar que una de las formas de controlar el acceso de los usuarios es a través de una variable de sesión que nos muestre el rol del usuario en curso. Esto sería un método manual el cual sería fácilmente controlable si hablamos de una aplicación pequeña y que sea diseñada por un solo desarrollador. Para aplicaciones de mediana y gran envergadura, existen las listas de control de acceso (ACL). Una lista de control de acceso o ACL (del inglés, *Access Control List*) es un concepto de seguridad informática usado para fomentar la separación de privilegios (Ballad & Ballad, 2008). Es una forma de determinar por medio de grupos y usuarios los permisos de acceso. En el caso de la Web, los permisos pueden determinar quién tiene privilegios de administrador para leer un documento, escribir en una base de datos, imprimir, etc.

Figura 5.1. *Ejemplo de ACL para usuarios basado en roles o perfiles*

La implementación de un ACL para el control de acceso a un servicio (página web) para restringir el acceso a los usuarios a páginas específicas de acuerdo a los permisos dados a una cuenta particular, es una labor que requiere de dos pasos:

1. La creación de un ACL.
2. Su utilización en la página web.

Para la creación del ACL podemos apoyarnos en el uso de de una base de datos que nos permita especificar la forma en la cual los permisos y roles son modelados en nuestro sistema. Teniendo en cuenta el modelo de roles establecido en la Figura 5.1, presentamos un pequeño esquema de base de datos que podría servir de orientación para la creación de un ACL. De acuerdo a las necesidades particulares de cada aplicación, los campos de cada tabla variarán. La estructura del ACL que proponemos se muestra en la Figura 5.2.

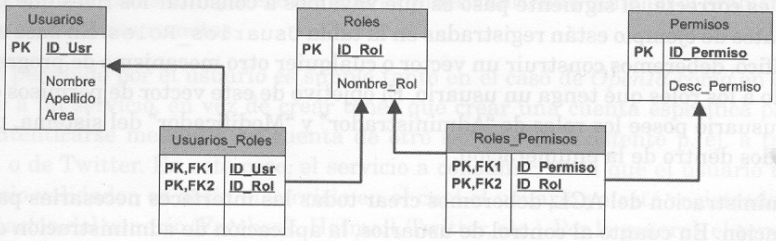

Figura 5.2. *Ejemplo de esquema ACL*

Una vez creado el ACL a través de una base de datos, el siguiente paso es la conexión de nuestras páginas web con dicho ACL. Como el ACL está implementado a través de una base de datos, este proceso será de la misma forma en la que se trata en cualquier aplicación de conexión a una base de datos. Así, de acuerdo al lenguaje de servidor que hayamos elegido para realizar la aplicación crearemos el código PHP o ASP que permita conectarse a una base de datos y para nuestro caso, elegiremos la base de datos ACL_ITESI. La forma de conectarse a una base de datos la especificamos en el capítulo 6 de este mismo libro. Hay que tener en cuenta que la implementación de un sistema completo de ACL cubre aspectos tanto de acceso y autenticación de la conexión de un cliente como de administración de los datos del ACL.

Para el manejo de acceso y autenticación del cliente debemos crear un servicio que se invocará cada vez que un usuario se autentique con la página web y, además, realice todas las operaciones pertinentes para dicho usuario. Es decir, una vez realizada la conexión con la base de datos que maneja el ACL, el siguiente paso es la creación y manejo de una clase ACL. Esta clase será la base para la definición de todos los conceptos ACL como el usuario en curso, los roles que posee un usuario en curso y, por último, los permisos que posee el usuario en curso. Un ejemplo de cómo puede ser creada esta clase sobre un lenguaje embebido como PHP es el siguiente:

```
<?php
    class ACL{
        var $perms = array();//Guarda los permisos de un usuario
        var $userID = 0; //Es el ID del usuario actual
        var $userRoles = array();//Guarda los roles del usuario
            actual

        function __construct($userID = ''){
            if ($userID != ''){
                $this->userID = floatval($userID);
            }else{
                $this->userID = floatval($_SESSION['userID']);
            }
            $this->userRoles = $this->getUserRoles('ids');
            $this->buildACL();
        }

        function ACL($userID='') {
            $this->__construct($userID);
        }
    }
?>
```

Si la autenticación es correcta, el siguiente paso es que vayamos a consultar los roles que posee el usuario y que para nuestra base de datos de ejemplo están registradas en la tabla `Usuarios_Roles`. En base a los roles encontrados para un usuario específico, deberemos construir un vector o cualquier otro mecanismo de programación para guardar los permisos de acuerdo a los roles que tenga un usuario. El objetivo de este vector de permisos es evitar repeticiones; así por ejemplo, si un usuario posee los roles de "Administrador" y "Modificador" del sistema, probablemente tendrá varios permisos repetidos dentro de la enumeración.

En cuanto a la administración del ACL, deberemos crear todas las interfaces necesarias para controlar todos los aspectos de administración. En cuanto al control de usuarios, la aplicación de administración del ACL debe permitir buscar a un usuario específico a través de alguna de las características registradas (nombre de usuario, nombre, email, etc.), permitir visualizar un listado con todos los usuarios registrados, donde sea posible visualizar los roles asignados a un usuario específico, visualizar los permisos de un usuario específico, editar los detalles de un usuario o editar los roles asignados a un usuario. De manera similar a la administración de usuarios, deberá existir una interfaz de manejo de roles que permita crear un nuevo rol y asignar los permisos a los que tiene acceso ese nuevo rol. Por último deberá existir una interfaz de administración de permisos donde de manera general se listen los permisos y se permita agregar, modificar o eliminar permisos.