Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Уфимский государственный нефтяной технический университет»

Кафедра вычислительной техники и инженерной кибернетики

Лабораторная работа №3

по дисциплине Информационная безопасность

на тему: «Реализация дискреционной модели политики безопасности»

Выполнил: студент гр. БПОи-16 К.И. Камалов

Проверил: доцент, кандидат наук каф. ВТИК Т.Х. Агишев

Уфа 2019

**Цель работы**: познакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

**Порядок выполнения работы**:

1. Выбрать язык программирования – JavaScript.

2. Из таблицы 1 получены следующие значения характеристик:

|  |  |  |
| --- | --- | --- |
| Вариант | Количество субъектов доступа (пользователей) | Количество объектов доступа |
| 24 | 10 | 5 |

Соответствие множеств типов доступов и соответствующих значений в матрице доступов будет следующее:

|  |  |  |
| --- | --- | --- |
| **Десятичное число** | **Двоичное число** | **Разрешенные типы доступов** |
| 0 | 000 | Полный запрет |
| 1 | 001 | Передача прав |
| 2 | 010 | Запись |
| 3 | 011 | Запись, Передача прав |
| 4 | 100 | Чтение |
| 5 | 101 | Чтение, Передача прав |
| 6 | 110 | Чтение, Запись |
| 7 | 111 | Полный доступ |

3. Написать программу-генератор паролей, в соответствие с требованиями Вашего варианта. Программа должна выполнять следующие действия:

function randomIntFromInterval(min,max) {

return Math.floor(Math.random()\*(max-min+1)+min);

}

function refreshTable() {

for (var i in Users) {

for (var j = 1; j <= 5; j++) {

document.getElementById(i).childNodes[j+1].innerHTML = rights[Users[i][j]];

}

}

}

function log\_out() {

active\_user = {};

action\_btn.style.visibility = "hidden";

actions\_list.style.visibility = "hidden";

active\_user\_name\_txt.style.visibility = "hidden";

}

function log\_in() {

alert("Вход совершен успешно");

action\_btn.style.visibility = "visible";

actions\_list.style.visibility = "visible";

active\_user\_name\_txt.style.visibility = "visible";

active\_user\_name\_txt.innerHTML = "Текущий пользователь: " + active\_user.name;

}

///////////////

// Data section

///////////////

var action\_btn = document.getElementById("action");

var actions\_list = document.getElementById("actions\_list");

var active\_user\_name\_txt = document.getElementById("active\_user\_name");

// 10 Users list

var Users = {

1: "Иван",

2: "Андрей",

3: "Александр",

4: "Рашид",

5: "Мария",

6: "Дмитрий",

7: "Ильшат",

8: "Вадим",

9: "Дарья",

10: "Эдуард",

};

Object.defineProperty(Users, "findUserByName", {enumerable: false, writable: true});

Users.findUserByName = function (name) {

for (var id in this) {

if (this.hasOwnProperty(id)) {

if (this[id].name == name) {

return +id;

}

}

}

};

// Rights list

var rights = {

0: "Полный запрет",

1: "Передача прав",

2: "Запись",

3: "Запись, передача прав",

4: "Чтение",

5: "Чтение, передача прав",

6: "Чтение, запись",

7: "Полный доступ",

}

var admin = {};

var active\_user = {};

/////////

// Begin

////////

// Hide interactions with user onload

log\_out();

// Random admin creation

admin.id = randomIntFromInterval(1, 10);

admin.name = Users[admin.id];

for (var i = 1; i < 6; i++) {

admin[i] = 7;

}

document.getElementById("admin\_name").innerHTML += Users[admin.id];

// Randomize rights

delete Users[admin.id];

document.getElementById(admin.id).insertCell(0).innerHTML = admin.name;

for (var j = 1; j <= 5; j++) {

document.getElementById(admin.id).insertCell(j).innerHTML = rights[7];

}

// Fill names and rights into table

var data = {};

for (var i in Users) {

data = {};

data.name = Users[i];

document.getElementById(i).insertCell(0).innerHTML = data.name;

for (var j = 1; j <= 5; j++) {

data[j] = randomIntFromInterval(0, 7);

document.getElementById(i).insertCell(j).innerHTML = rights[data[j]];

}

Users[i] = data;

}

///////////////////

// Event listeners

///////////////////

// User authorization

document.getElementById("sign\_in").addEventListener("click", function () {

input\_name = prompt("Введите идентификатор");

for (var i in Users) {

if (Users[i].name == input\_name) {

active\_user = Users[i];

log\_in();

return;

}

}

if (input\_name == admin.name) {

active\_user = admin;

log\_in();

} else {

alert("Неверный ввод, попробуйте еще раз");

log\_out();

}

});

// User actions

var action;

var file, rightTo, give\_name;

action\_btn.addEventListener("click", function () {

action = +prompt("Введите номер действия");

if (action == 3) {

file = +prompt("Право на какой объект передается?");

if (file <= 0 && file > 6) {

alert("Выбран неверный файл");

return;

}

rightTo = +prompt("Какое право передается?", "2 - запись или 4 - чтение");

if (rightTo != 2 && rightTo != 4) {

alert("Выбрано неверное право");

return;

} else {

if (active\_user[file] - rightTo != 1 &&

active\_user[file] != 7) {

alert("Вы не можете передать данное право");

return;

}

}

give\_name = prompt("Идентификатор получателя");

var give\_id = Users.findUserByName(give\_name);

if (rightTo == Users[give\_id][file] ||

rightTo == Users[give\_id][file] - 1 ||

Users[give\_id][file] == 7 ||

Users[give\_id][file] == 6) {

alert("У данного пользователя уже есть данное право");

return;

} else {

Users[give\_id][file] += rightTo;

alert("Передача прав успешно совершена");

refreshTable();

}

}

if (action == 2) {

file = +prompt("В какой файл производить запись?");

if (file <= 0 && file > 6) {

alert("Выбран неверный файл");

return;

}

if (active\_user[file] != 2 &&

active\_user[file] != 3 &&

active\_user[file] != 6 &&

active\_user[file] != 7) {

alert("Вы не можете производить запись в данный файл");

return;

}

alert("Запись успешно произведена");

}

if (action == 1) {

file = +prompt("Какой файл читать?");

if (file <= 0 && file > 6) {

alert("Выбран неверный файл");

return;

}

if (active\_user[file] != 4 &&

active\_user[file] != 5 &&

active\_user[file] != 6 &&

active\_user[file] != 7) {

alert("Вы не можете читать данный файл");

return;

}

alert("Чтение успешно завершено");

}

if (action == 4) {

log\_out();

}

});

Скриншот реализации



**Ответы на контрольные вопросы:**

1. Что понимается под политикой безопасности в компьютерной системе?

Под политикой безопасности понимают набор норм, правил и практических приемов, которые регулируют управление, защиту и распределение ценной информации. Политика безопасности задает механизмы управления доступа к объекту, определяет, как разрешенные, так и запрещенные доступы.

1. В чем заключается модель дискреционной политики безопасности в компьютерной системе?

Дискреционная политика определяет отображение объектов на пользователей-субъектов. В соответствии с данным отображением, каждый объект объявляется собственностью соответствующего пользователя, который может выполнять над ними определенную совокупность действий, в которую могут входить несколько элементарных действий (чтение, запись, модификация и т.д.). Пользователь, являющийся собственником объекта, иногда имеет право передавать часть или все права другим пользователям (обладание администраторскими правами).

1. Что понимается под матрицей доступа в дискреционной политике безопасности? Что хранится в данной матрице?

Указанные права доступа пользователей-субъектов к объектам компьютерной системы записываются в виде так называемой МАТРИЦЫ ДОСТУПА. На пересечении i-ой строки и j-ого столбца данной матрицы располагается элемент Sij – множество разрешенных действий j-ого пользователя над i-ым объектом.

1. Какие действия производятся над матрицей доступа в том случае, когда один субъект передает другому субъекту свои права доступа к объекту компьютерной системы?

Пользователь, обладающий правами передачи своих прав доступа к объекту другому пользователю, может сделать это. При этом, пользователь, передающий права, может указать непосредственно, какие из своих прав он передает другому.

**Выводы по работе:**

В ходе выполнения данной лабораторной работы была реализована дискреционная модель политики безопасности, отвечающая заданным требованиям.