Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

ЛАБОРАТОРНАЯ

РАБОТА №4

По Информационная безопасность

Тема Реализация дискреционной модели политики безопасности.

Руководитель

Астафьев А.В.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

Лабораторная работа №4

Тема: реализация дискреционной модели политики безопасности.

Цели и задачи: познакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

Ход работы: задание: реализовать программный модуль, создающий матрицу доступа пользователей к объектам компьютерной системы:

Листинг кода 1 – Form1.cs

using System;

using System.Windows.Forms;

namespace LR4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

cbObject.DataSource = accessObjectsStrings;

SingleRight[] rights = new SingleRight[]

{

new SingleRight(right.None.ToString(),right.None) ,

new SingleRight(right.Client.ToString(),right.Client),

new SingleRight(right.Employer.ToString(),right.Employer),

new SingleRight(right.Admininstrator.ToString(), right.Admininstrator)

};

cbRight.DataSource = rights;

cbRight.DisplayMember = "Name";

}

string[] accessObjectsStrings = new string[] {

"Окно работы клиента",

"Окно работы сотрудника",

"Окно работы администратора"

//"Web-сервер"

};

public class SingleRight

{

public string Name { get; set; }

public right Right { get; set; }

public SingleRight(string name, right right)

{

Name = name;

Right = right;

}

}

public class accessRights

{

public right clientForm;

public right employerForm;

public right adminForm;

}

// Объявим перечисление для хранения прав доступа

public enum right

{

None,

Client,

Employer,

//Moderator,

Admininstrator

}

// Объявим структуру для хранения данных пользователя

public class User

{

public string Name { get; set; }

public accessRights rights;

}

// Переменная для хранения количества пользователей

int usersCount;

// Список для хранения данных о пользователе

User[] listOfUsers;

// Объявим экземпляр класса Random для генерации прав

Random r = new Random();

private void lbUsers\_SelectedIndexChanged(object sender, EventArgs e)

{

dgwUsersRightsRefresh();

}

private void bUserGenerate\_Click(object sender, EventArgs e)

{

// Получаем количество пользователей

usersCount = Convert.ToInt32(tbUserCount.Text);

// Очищаем ListBox

lbUsers.Items.Clear();

// Выделяем памать под статический массив пользователей

listOfUsers = new User[usersCount + 1]; // +1 т.к. Администратор отдельно

// В цикле заполняем массив пользователей

for (int i = 0; i < usersCount; i++)

{

listOfUsers[i] = new User();

listOfUsers[i].Name = $"Пользователь № {i + 1}";

listOfUsers[i].rights = new accessRights();

listOfUsers[i].rights.clientForm = (right)r.Next(0, 4);

listOfUsers[i].rights.employerForm = (right)r.Next(0, 4);

listOfUsers[i].rights.adminForm = (right)r.Next(0, 4);

}

listOfUsers[usersCount] = new User();

// Отдельно добавляем администратора

listOfUsers[usersCount].Name = "Администратор";

listOfUsers[usersCount].rights = new accessRights();

listOfUsers[usersCount].rights.clientForm = right.Admininstrator;

listOfUsers[usersCount].rights.employerForm = right.Admininstrator;

listOfUsers[usersCount].rights.adminForm = right.Admininstrator;

// В цикле выводим список пользователей

for (int i = 0; i < usersCount + 1; i++)

{

lbUsers.Items.Add(listOfUsers[i].Name);

}

cbUser.DataSource = listOfUsers;

cbUser.DisplayMember = "Name";

}

private void bChangeUserRights\_Click(object sender, EventArgs e)

{

User user = cbUser.SelectedItem as User;

string currentObject = cbObject.SelectedItem as string;

switch (currentObject)

{

case "Окно работы клиента":

user.rights.clientForm = getCbRightsValue();

break;

case "Окно работы сотрудника":

user.rights.employerForm = getCbRightsValue();

break;

case "Окно работы администратора":

user.rights.adminForm = getCbRightsValue();

break;

}

dgwCurrentUserRefresh();

dgwUsersRightsRefresh();

}

private right getCbRightsValue()

{

SingleRight right = cbRight.SelectedItem as SingleRight;

return right.Right;

}

private void cbUser\_SelectedIndexChanged(object sender, EventArgs e)

{

dgwCurrentUserRefresh();

}

private void dgwUsersRightsRefresh()

{

// Очищаем компонент dataGridView

dgwUsersRights.Rows.Clear();

// Выводим персональные данные выбранного пользователя

dgwUsersRights.Rows.Add(accessObjectsStrings[0], listOfUsers[lbUsers.SelectedIndex].rights.clientForm.ToString());

dgwUsersRights.Rows.Add(accessObjectsStrings[1], listOfUsers[lbUsers.SelectedIndex].rights.employerForm.ToString());

dgwUsersRights.Rows.Add(accessObjectsStrings[2], listOfUsers[lbUsers.SelectedIndex].rights.adminForm.ToString());

}

private void dgwCurrentUserRefresh()

{

dgwCurrentUser.Rows.Clear();

User user = cbUser.SelectedItem as User;

// Выводим персональные данные выбранного пользователя

dgwCurrentUser.Rows.Add(accessObjectsStrings[0], user.rights.clientForm.ToString());

dgwCurrentUser.Rows.Add(accessObjectsStrings[1], user.rights.employerForm.ToString());

dgwCurrentUser.Rows.Add(accessObjectsStrings[2], user.rights.adminForm.ToString());

}

}

}

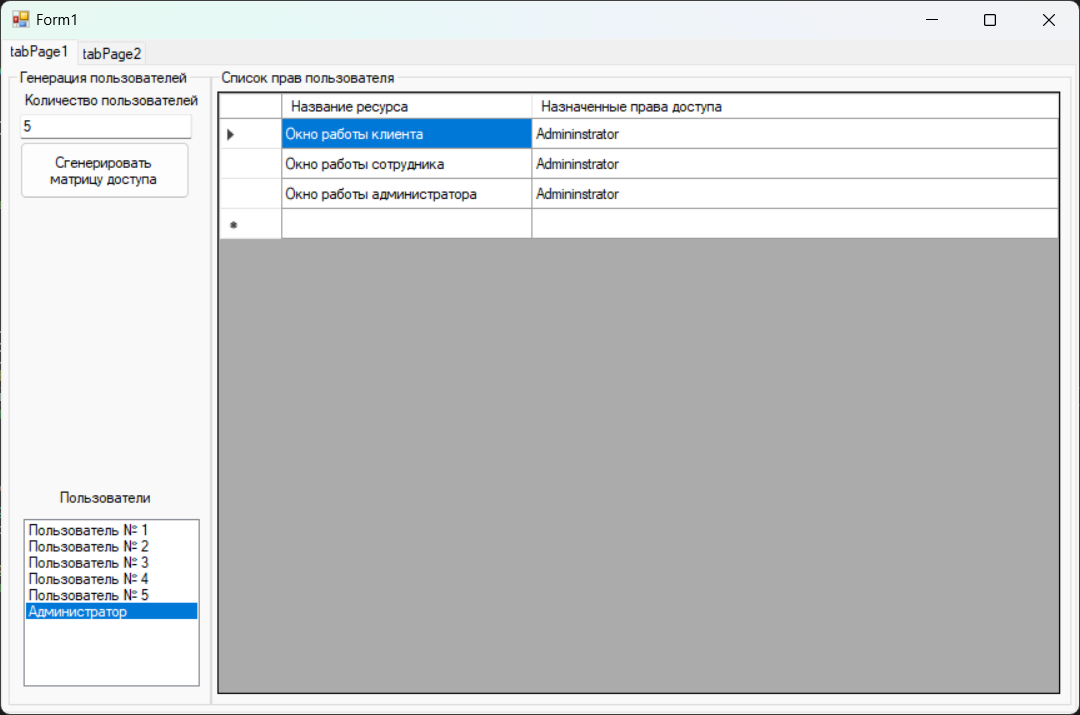


Рисунок 1 – Создание пользователей и просмотр прав

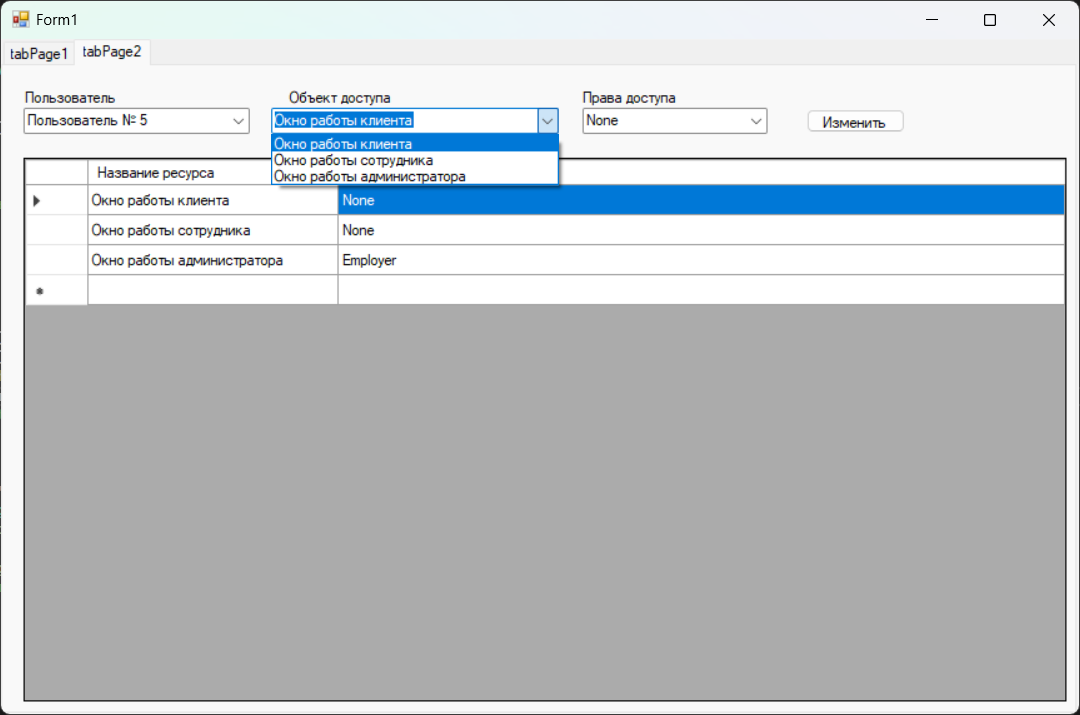


Рисунок 2 – Выбор объекта доступа для изменения

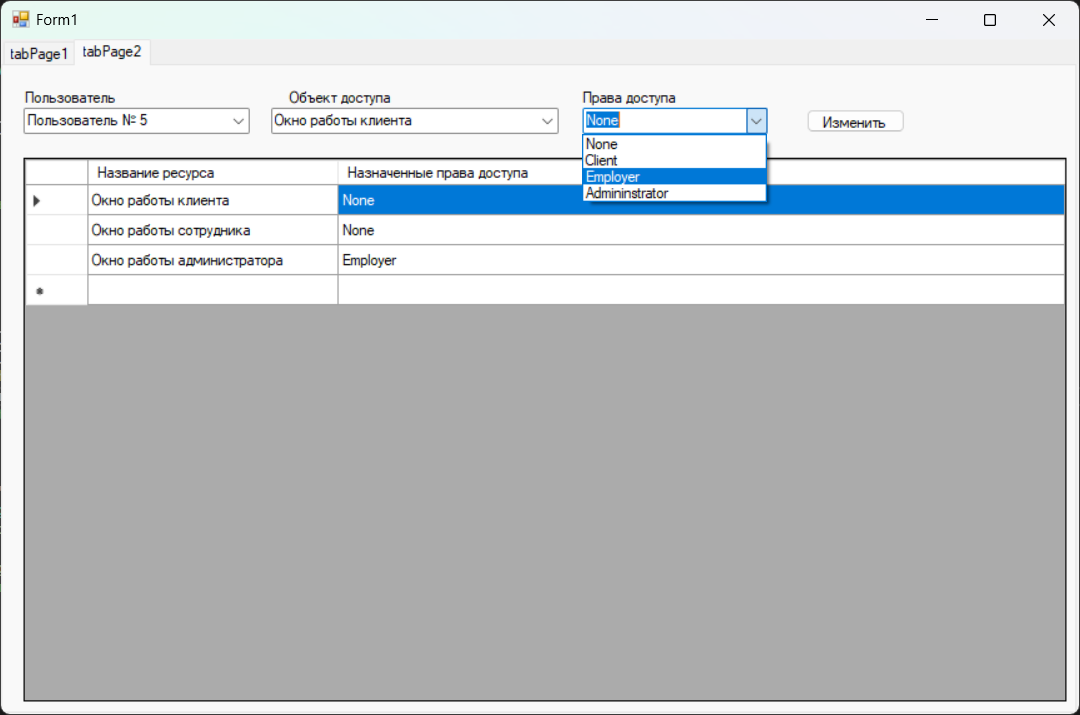


Рисунок 3 – Назначение права доступа

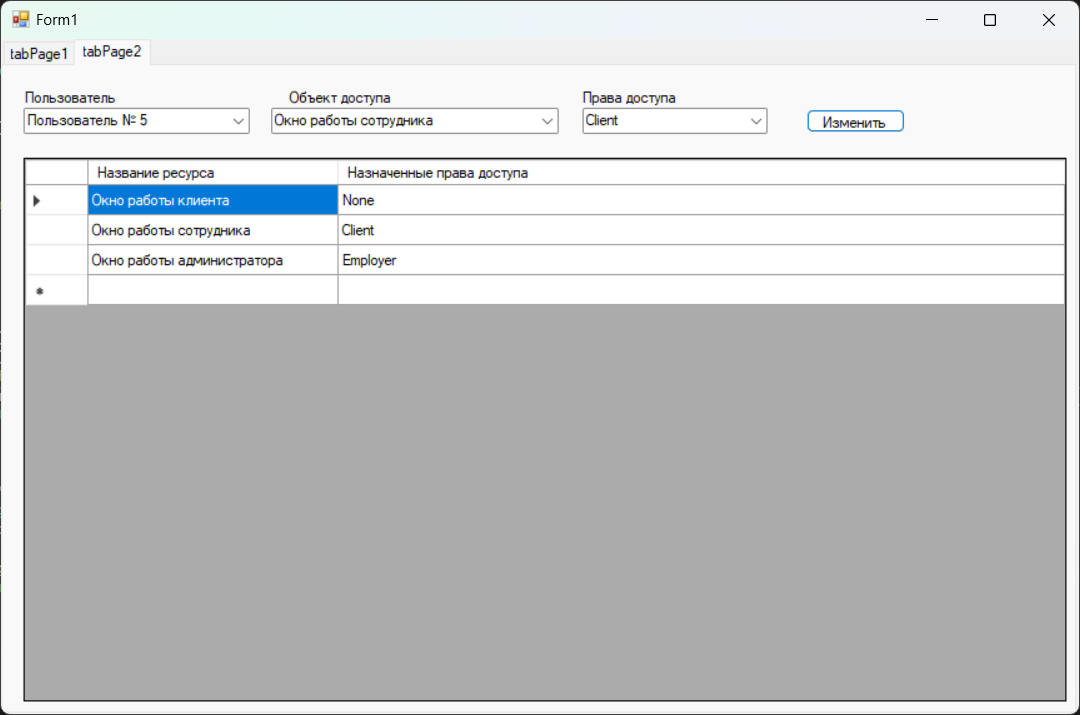


Рисунок 4 – Демонстрация изменения права доступа

Вывод: в ходе работы познакомились с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.