Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Лабораторная работа № 5

**«Изучение механизмов защиты ОС»**

## Отчёт по дисциплине

**«Методы защиты КС»**

Пенза 2020

**Выполнили с-ты гр. 17ВВ2:**

Живаев В.  
Безжонов Р.

**Приняли:**

к.т.н. Дубравин А.В.  
Кармышева Н.С.

**Цель работы.**

Научиться пользоваться функциями системного API, предоставляющего доступ к настройке дескрипторов безопасности отдельных компонентов.

**Задание.**

Разработать программу, выполняющую создание записи в реестре с правами «полный доступ» для администратора и только чтение для текущего пользователя.

**Листинг.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdafx.h"

#include<stdio.h>

#include<windows.h>

#include<AccCtrl.h>

#include<Aclapi.h>

#include<tchar.h>

#ifdef \_DEBUG

#include<conio.h>

#endif

void error(const char \*error) {

printf("%s\n", error);

#ifdef \_DEBUG

\_getch();

#endif

exit(1);

}

int main(int argc, char\* argv[]) {

EXPLICIT\_ACCESS ea[2];

ZeroMemory(&ea, 2 \* sizeof(EXPLICIT\_ACCESS));

//ïîëó÷åíèåèäåíòèôèêàòîðàãðóïïû Everyone

SID\_IDENTIFIER\_AUTHORITY SIDAuthLocal = SECURITY\_CREATOR\_SID\_AUTHORITY;

PSID pLocalUserSID = NULL;

if(!AllocateAndInitializeSid(&SIDAuthLocal, 1, SECURITY\_CREATOR\_OWNER\_RID, 0, 0, 0, 0, 0, 0, 0, &pLocalUserSID)) {

error("AllocateAndInitializeSid Error");

}

ea[0].grfAccessPermissions = KEY\_READ;

ea[0].grfAccessMode = SET\_ACCESS;

ea[0].grfInheritance = NO\_INHERITANCE;

ea[0].Trustee.TrusteeForm = TRUSTEE\_IS\_SID;

ea[0].Trustee.TrusteeType = TRUSTEE\_IS\_WELL\_KNOWN\_GROUP;

ea[0].Trustee.ptstrName = (LPTSTR) pLocalUserSID;

//ïîëó÷åíèåèäåíòèôèêàòîðàãðóïïû Admin

SID\_IDENTIFIER\_AUTHORITY SIDAuthNT = SECURITY\_NT\_AUTHORITY;

PSID pAdminSID = NULL;

if(!AllocateAndInitializeSid(&SIDAuthNT, 2, SECURITY\_BUILTIN\_DOMAIN\_RID, DOMAIN\_ALIAS\_RID\_ADMINS, 0, 0, 0, 0, 0, 0, &pAdminSID)) {

FreeSid(pLocalUserSID);

error("AllocateAndInitializeSid Error");

}

ea[1].grfAccessPermissions = KEY\_ALL\_ACCESS;

ea[1].grfAccessMode = SET\_ACCESS;

ea[1].grfInheritance = NO\_INHERITANCE;

ea[1].Trustee.TrusteeForm = TRUSTEE\_IS\_SID;

ea[1].Trustee.TrusteeType = TRUSTEE\_IS\_GROUP;

ea[1].Trustee.ptstrName = (LPTSTR) pAdminSID;

PACL pACL = NULL;

DWORD dwRes = SetEntriesInAcl(2, ea, NULL, &pACL);

if(dwRes != ERROR\_SUCCESS) {

FreeSid(pLocalUserSID);

FreeSid(pAdminSID);

error("SetEntriesInAcl Error");

}

PSECURITY\_DESCRIPTOR pSD = (PSECURITY\_DESCRIPTOR) LocalAlloc(LPTR, SECURITY\_DESCRIPTOR\_MIN\_LENGTH);

if(pSD == NULL) {

FreeSid(pLocalUserSID);

FreeSid(pAdminSID);

LocalFree(pACL);

error("LocalAlloc Error");

}

if(!InitializeSecurityDescriptor(pSD, SECURITY\_DESCRIPTOR\_REVISION)) {

FreeSid(pLocalUserSID);

FreeSid(pAdminSID);

LocalFree(pACL);

LocalFree(pSD);

error("InitializeSecurityDescriptor Error");

}

if(!SetSecurityDescriptorDacl(pSD, TRUE, pACL, FALSE)) {

FreeSid(pLocalUserSID);

FreeSid(pAdminSID);

LocalFree(pACL);

LocalFree(pSD);

error("SetSecurityDescriptorDacl Error");

}

SECURITY\_ATTRIBUTES sa;

sa.nLength = sizeof(SECURITY\_ATTRIBUTES);

sa.lpSecurityDescriptor = pSD;

sa.bInheritHandle = FALSE;

HKEY hkSub;

DWORD dwDisposition;

LONG lRes = RegCreateKeyEx(HKEY\_CURRENT\_USER, \_T("Software\\pi"), 0, NULL, 0, KEY\_READ | KEY\_WRITE, &sa, &hkSub, NULL);

printf("RegCreateKeyEx result: ");

switch(lRes) {

case ERROR\_SUCCESS:

printf("OK!\n");

break;

default:

printf("ERROR!\n");

}

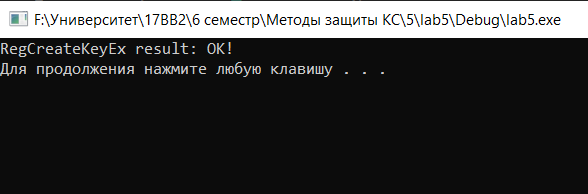
system("pause");

printf("\nFinished!");

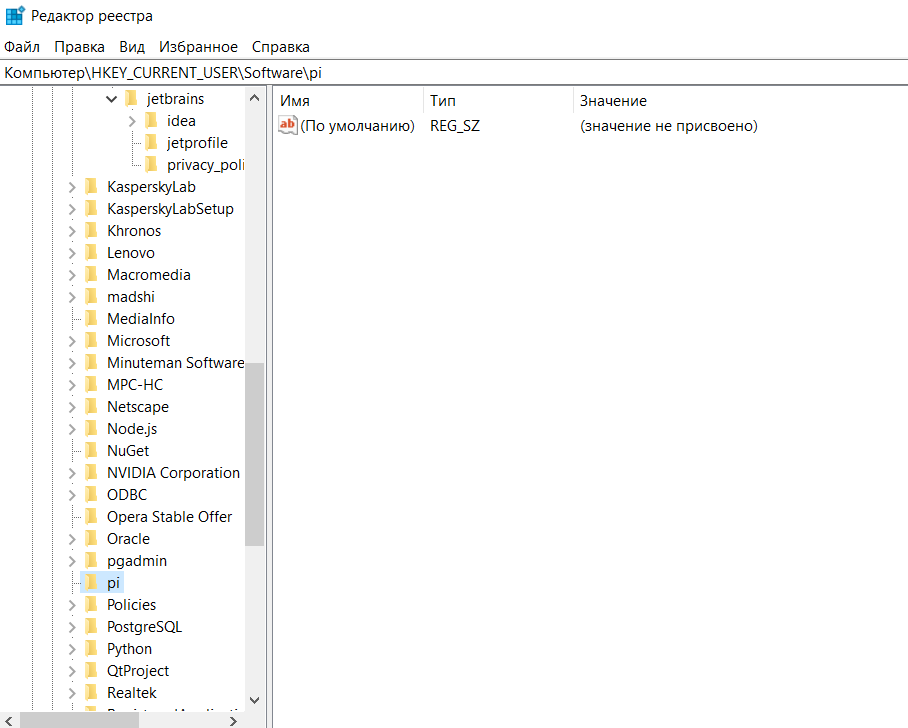
}

**Результат работы.**

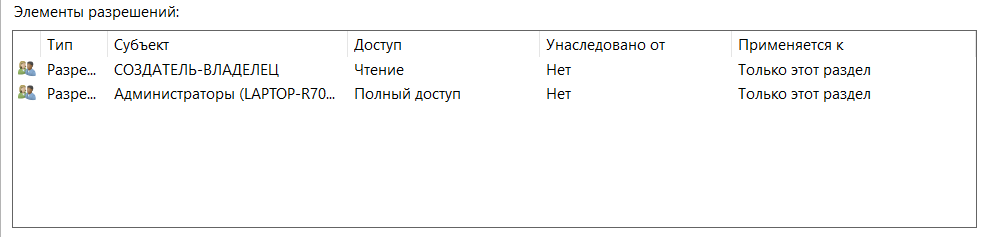
Результат работы программы показан на рисунках 1-5.



**Рисунок 1 – результат в консоли.**



**Рисунок 2 – Новый каталог в реестре.**



**Рисунок 3 – Политики доступа к элементу реестра.**

**Вывод.**

Научились работать с функциями системного API для настройки дескрипторов безопасности отдельных элементов. Изучили структуры дескриптора безопасности и прав доступа.