Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Защита информации в сети Internet»

на тему «Атрибуты безопасности в ОС Windows»

Выполнили  
студенты группы 18ВВ1:

Арсентьева У.С.  
Горбунова В.В.

Приняли:

к.т.н., доцент Дубравин А.В.

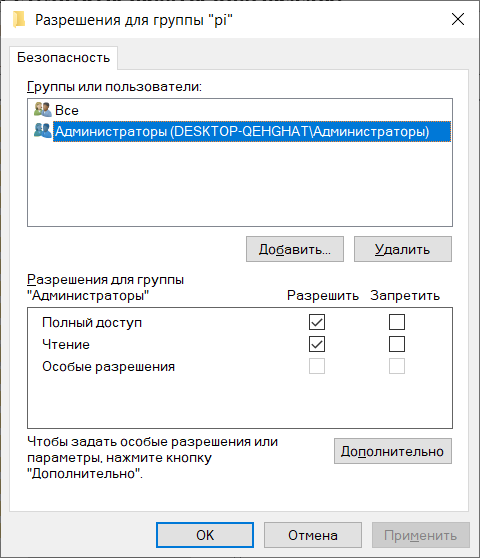
к.т.н., доцент Карамышева Н.С.

2020

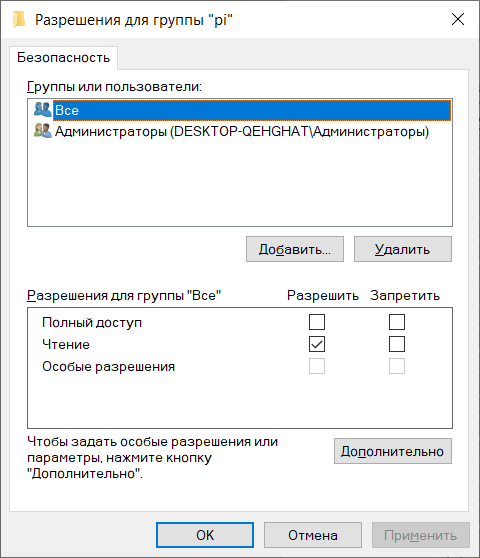
**Цель работы:** Изучить методы работы с атрибутами безопасности в ОС Windows.

**Задание:** Разработать программу, выполняющую создание раздела реестра «HKEY\_CURRENT\_USER\Software\pi» с атрибутами безопасности для группы пользователей «Администраторы» и правами на полный доступ, а для группы «Все» только чтение.

**Результаты работы программы:**



**Рисунок 1 – Разрешения для администраторов**



**Рисунок 2 – Разрешения для всех**

**Вывод:** Была разработана программа, выполняющая внедрение и извлечение некоторых данных из файла с изображением.

**Листинг программы:**

#include <stdio.h>

#include <conio.h>

#include <windows.h>

#include <winuser.h>

#include <iostream>

#include <accctrl.h>

#include <aclapi.h>

void main()

{

wchar\_t mus[] = { L"" };

DWORD dwRes;

PSID pEveryoneSID = NULL, pAdminSID = NULL;

PACL pACL = NULL;

PSECURITY\_DESCRIPTOR pSD = NULL;

HKEY hkSub = NULL;

SID\_IDENTIFIER\_AUTHORITY SIDAuthWorld = SECURITY\_WORLD\_SID\_AUTHORITY;

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

if (!AllocateAndInitializeSid(&SIDAuthWorld, 1,

SECURITY\_WORLD\_RID, 0, 0, 0, 0, 0, 0, 0,

&pEveryoneSID))

{

wprintf(L"AllocateAndInitializeSid Error %u\n", GetLastError());

goto Cleanup;

}

if (!AllocateAndInitializeSid(&SIDAuthNT, 2,

SECURITY\_BUILTIN\_DOMAIN\_RID,

DOMAIN\_ALIAS\_RID\_ADMINS,

0, 0, 0, 0, 0, 0,

&pAdminSID))

{

wprintf(L"AllocateAndInitializeSid Error %u\n", GetLastError());

goto Cleanup;

}

EXPLICIT\_ACCESS ea[2];

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

ea[0].grfAccessPermissions = GENERIC\_READ;

ea[0].grfAccessMode = SET\_ACCESS;

ea[0].grfInheritance = CONTAINER\_INHERIT\_ACE;

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

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

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

ea[1].grfAccessPermissions = GENERIC\_ALL;

ea[1].grfAccessMode = SET\_ACCESS;

ea[1].grfInheritance = CONTAINER\_INHERIT\_ACE;

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

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

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

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

if (ERROR\_SUCCESS != dwRes)

{

wprintf(L"SetEntriesInAcl Error %u\n", GetLastError());

goto Cleanup;

}

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

if (NULL == pSD)

{

wprintf(L"LocalAlloc Error %u\n", GetLastError());

goto Cleanup;

}

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

{

wprintf(L"InitializeSecurityDescriptor Error %u\n", GetLastError());

goto Cleanup;

}

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

{

wprintf(L"SetSecurityDescriptorDacl Error %u\n", GetLastError());

goto Cleanup;

}

SECURITY\_ATTRIBUTES sa;

sa.nLength = sizeof(SECURITY\_ATTRIBUTES);

sa.lpSecurityDescriptor = pSD;

sa.bInheritHandle = FALSE;

if (RegOpenKey(HKEY\_CURRENT\_USER, L"SOFTWARE\\", &hkSub) == 0) {

RegDeleteKey(hkSub, L"test29");

RegCloseKey(hkSub);

}

wprintf(L"RegCreateKeyEx result %u\n",

RegCreateKeyEx(HKEY\_CURRENT\_USER, L"SOFTWARE\\pi", 0, mus,

REG\_OPTION\_NON\_VOLATILE, KEY\_SET\_VALUE, &sa, &hkSub, &dwRes));

if (hkSub) {

char szPath[] = "123";

if (RegSetValueExA(hkSub, "Password", 0, REG\_SZ, (LPBYTE)szPath, strlen(szPath)) == ERROR\_SUCCESS)

{

MessageBoxW(NULL, L"Key was created!", L"Success", MB\_OK);

}

else

MessageBoxW(NULL, L"Key not was created!", L"Error", MB\_OK);

}

Cleanup:

if (pEveryoneSID)

FreeSid(pEveryoneSID);

if (pAdminSID)

FreeSid(pAdminSID);

if (pACL)

LocalFree(pACL);

if (pSD)

LocalFree(pSD);

if (hkSub)

RegCloseKey(hkSub);

system("regedit");

return;

}