

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЕТ
к лабораторной работе № 6
на тему

Защита ПО от несанкционированного использования

Выполнил

Тимофеев К.А.

Проверил

Е.А. Лещенко

Минск 2024

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение..... | 3 |
| 1 Краткие теоретические сведения..... | 4 |
| 2 Результаты выполнения лабораторной работы..... | 6 |
| Заключение | 10 |
| Приложение А (обязательное) Листинг кода..... | 11 |

ВВЕДЕНИЕ

Цель данной лабораторной работы заключается в ознакомлении с основными технологиями защиты программного обеспечения от несанкционированного использования, получении навыков защиты разработанной программы от несанкционированного копирования.

Необходимо реализовать на выбор 3 метода обфускации программного кода приложения, разработанного в рамках лабораторных работ 4-5, позволяющие защитить ПО от несанкционированного использования в следующих комбинациях: по одному, любые 2 на выбор из трех одновременно, все три одновременно.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Обфускация или запутывание кода – это приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

Обфускация производится в следующих целях:

1 Затруднение декомпиляции/отладки и изучения программ с целью обнаружения функциональности.

2 Затруднение декомпиляции проприетарных программ с целью предотвращения обратной разработки или обхода DRM и систем проверки лицензий.

3 Оптимизация программы с целью уменьшения размера работающего кода и (если используется некомпилируемый язык) ускорения работы.

4 Демонстрация неочевидных возможностей языка и квалификации программиста (если производится вручную, а не инструментальными средствами).

«Запутывание» кода может осуществляться на уровне алгоритма, исходного текста и/или ассемблерного текста. Для создания запутанного ассемблерного текста могут использоваться специализированные компиляторы, использующие неочевидные или недокументированные возможности среды исполнения программы. Существуют также специальные программы, производящие обфускацию, называемые обфускаторами (англ. obfuscator).

Наиболее известными и частыми являются следующие виды обфускации:

1 Лексическая обфускация.

Это самый простой вид обфускации, при котором происходит удаление комментариев или замена их на бессмысленные, удаление символов форматирования кода, замена имён на трудно воспринимаемые (часто очень похожие последовательности нулей и букв О, единиц и букв I и т. д.), изменение расположение частей программы, добавление лишних операций, которые не повлияют на итоговый результат выполнения.

2 Обфускация структур данных.

Преобразования данного вида связаны с трансформацией структур данных. Существует следующие методы преобразования данных: обфускация хранения (предполагает создание и использование нетривиальных типов данных, замена существующих типов данных и т.п., используются следующие

методы: разделение одной переменной на комбинацию переменных, преобразование статистических данных в процедурные, изменение области действия переменной, изменение интерпретации данных); обфускация соединения (состоит в соединении нескольких зависимых данных или разделении независимых данных, например, массив можно разделить на несколько или наоборот несколько массивов соединить в один); обфускация переупорядочивания (включает в себя изменение порядка объявления переменных или функций, изменение внутреннего порядка структур данных).

3 Обфускация потока управления.

Данный вид преобразований направлен на изменения порядка выполнения блоков программы. Включает в себя следующие методы: вставка функций (вставка тела функции вместо вызова функции), извлечение функции (выделение участка кода в отдельную функцию), распараллеливание кода в тех участках, где это возможно осуществить, добавление «мёртвого» кода, добавление недостижимого кода, добавление избыточного кода, устранение библиотечных вызовов, преобразование графа потока управления, клонирование функций, объединение функций, реструктуризация циклов, расширение области действия переменной, переупорядочивание циклов, выражений, условных и безусловных переходов, вызовов функций.

2 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В результате разработки программы были реализованы на выбор 3 метода обфускации программного кода приложения в следующих комбинациях: по одному (рисунки 2.1-2.3), любые 2 на выбор из трех одновременно (рисунок 2.4), все три одновременно (рисунок 2.5).

```
public partial class Form1 : Form
{
    Ссылка: 5
    private enum R
    {
        A,
        U,
        G
    }

    Ссылка: 4
    private class D
    {
        Ссылка: 2
        public bool P { get; set; }

        public bool B { get; set; }
        Ссылка: 2
        public bool X { get; set; }
        Ссылка: 3
        public bool Ds { get; set; }
    }
}
```

Рисунок 2.1 – Использование лексической обфускации

```

Ссылка: 0
public Form3()
{
    InitializeComponent();

    usersData.Add(new User("a", "1"));
    InitializeUsersData();
    usersData[0].Role = 0;
    usersData[1].Role = Roles.Guest;
    usersData[2].Role = Roles.Admin;
    AssignControlsToUsers();
    Users();
}

Ссылка: 1
private void InitializeUsersData()
{
    usersData.Add(new User("u", "1"));
    usersData.Add(new User("g", "1"));
}

Ссылка: 1
private void Users()
{
    usersData[0].Role = Roles.Admin;
    usersData[1].Role = Roles.User;
    usersData[2].Role = Roles.Guest;
}

```

Рисунок 2.2 – Использование обфускации потока управления

```

private class User
{
    private P1 a = new();
    private P2 b = new();
    Ссылка: 6
    public string Login { get => a.Username; set => a.Username = value; }
    Ссылка: 2
    public string Password { get => b.Password; set => b.Password = value; }
    Ссылка: 5
    public int Role { get => a.Role; set => a.Role = value; }
    Ссылка: 5
    public bool IsSignedIn { get => a.IsSignedIn; set => a.IsSignedIn = value; }
    Ссылка: 5
    public int SlotIndex { get => b.SlotIndex; set => b.SlotIndex = value; }

    public User(string login, string password)
    {
        Login = login;
        Password = password;
    }

    public override bool Equals(object obj)
    {
        return obj is User user && Login == user.Login;
    }
}

```

Рисунок 2.3 – Использование обфускации структур данных

```

Ссылка: 7
private class n9zyErY2wh
{
    private An1 a = new An1();
    private B4V3 b = new B4V3();

    private float ch = 39;
    Ссылка: 6
    public string NJwgJuvx1Z
    {
        get
        {
            return a.JHG5asdf;
        }
        set
        {
            a.JHG5asdf = value;
        }
    }

    Ссылка: 2
    public string MZHOzndTyj
    {
        get
        {
            return b.HG7dkjh2h;
        }
        set
        {
            b.HG7dkjh2h = value;
        }
    }
}

```

Рисунок 2.4 – Использование первых двух методов одновременно


```

public Form123()
{
    InitializeComponent();

    zWUtEktC4i.Add(new n9zyErY2wh("a", "1"));
    mGCWpEDIrH();
    zWUtEktC4i[0].fdMgppHkvd = -5463456;
    zWUtEktC4i[1].fdMgppHkvd = 223423;
    zWUtEktC4i[2].fdMgppHkvd = 654;
    gjTp20piko();
    FGetsdfrt();
}

Ссылка: 1
private void FGetsdfrt()
{
    zWUtEktC4i[0].fdMgppHkvd = 0;
    zWUtEktC4i[1].fdMgppHkvd = 1;
    zWUtEktC4i[2].fdMgppHkvd = 2;
}

Ссылка: 1
private void mGCWpEDIrH()
{
    zWUtEktC4i.Add(new n9zyErY2wh("u", "1"));
    zWUtEktC4i.Add(new n9zyErY2wh("g", "1"));
}

Ссылка: 1
private void gjTp20piko()
{
    for (int i = 1; i <= zWUtEktC4i.Count; i++)
    {
        T0LTxj6iD3xZYdp.Add(this.Controls[$"Message{i}"] as TextBox);
        ncZVDbRrWgfleAp.Add(this.Controls[$"SignOut{i}"] as Button);
        Cz1vwHqHigGg7K.Add(this.Controls[$"Send{i}"] as Button);
        xvcX7GqZw3PQir0.Add(this.Controls[$"Username{i}"] as Label);
    }
}

```

Рисунок 2.5 – Использование всех трех методов одновременно

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной лабораторной работы были реализованы на выбор 3 метода обфускации программного кода приложения, разработанного в рамках лабораторных работ 4-5, позволяющие защитить ПО от несанкционированного использования в следующих комбинациях: по одному, любые 2 на выбор из трех одновременно, все три одновременно.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг кода

```
using System.Text;
using System.Security.Cryptography;
using System.Windows.Forms;
using System.Linq;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
using Microsoft.VisualBasic.ApplicationServices;
using System.Security.Permissions;

namespace Lab6
{
    public partial class Form123 : Form
    {
        private int POxGI0R4gJ = 2;

        private enum bpiMKeviZb
        {
            bpiMKeviZ8,
            bpiMKevizb
        }

        private class y9RlJO2oIx
        {
            public bool bpiMKevlZb { get; set; }
            public bool bpiMKevlzb { get; set; }
            public bool bpiMKevlz8 { get; set; }
            public bool bpliMKevlzb { get; set; }
        }

        public class An1
        {
            static public int bpiMKevl sdfzb = 39;
            public string JHG5asdf { get; set; }
            public int swer5645654ldfjhSRFGsdlfkjGkljhHlk41 { get; set; }
            public bool IsSignedIn { get; set; }
        }

        public class B4V3
        {
            public string HG7dkjh2h { get; set; }

            private int sdsGDfg5gjhJH;
            public int g245GJHJKG8
            {
                get
                {
                    var cst = DFq435hk54jFHDlk235(Ff7Fflkjha3kajsdf8, 4);
                    var a = cst ^ 7;
                }
            }
        }
    }
}
```

```

        return sdsGDfg5gjhJH + ((a | Ff7Fflkjha3kajsdf8 / 6) &
~(a & Ff7Fflkjha3kajsdf8 / 7 + 1));
    }
    set
    {
        sdsGDfg5gjhJH = value - DFq435hk54jFHDlk235();
    }
}

private int Ff7Fflkjha3kajsdf8 = 42;

private int DFq435hk54jFHDlk235(int a = 3, int c = 4)
{
    Random aa = new Random();

    var b = (aa.Next(c) + a / 3 * 52) % 4;

    int z = Convert.ToInt32((b + 42 * Math.Pow(10, b)) / Math.PI
* (70 - (Ff7Fflkjha3kajsdf8 + 28)));

    return (int)(An1.bpiMkevlsdfzb + z + 52);
}
}

private class n9zyErY2wh
{
    private An1 a = new An1();
    private B4V3 b = new B4V3();

    private float ch = 39;
    public string NJwgJuvx1Z
    {
        get
        {
            return a.JHG5asdf;
        }
        set
        {
            a.JHG5asdf = value;
        }
    }

    public string MZHOzndTyj
    {
        get
        {
            return b.HG7dkjh2h;
        }
        set
        {
            b.HG7dkjh2h = value;
        }
    }
}

```

```

        public int fdMgppHkvd { get { return
a.swer5645654ldfjhSRFGsdlfkjGkljhHlk41; } set {
a.swer5645654ldfjhSRFGsdlfkjGkljhHlk41 = value; } }
        public bool GzZaKKFzHq { get; set; }

        public int l0U5EAa3NZ
        {
            get
            {
                return b.g245GJHJKG8;
            }
            set
            {
                b.g245GJHJKG8 = value;
            }
        }

        public n9zyErY2wh(string login, string password)
        {
            NJwgJuvx1Z = login;
            MZHOzndTyj = password;
        }

        public override bool Equals(object M15rWQazOm)
        {
            return M15rWQazOm is n9zyErY2wh user && NJwgJuvx1Z ==
user.NJwgJuvx1Z;
        }
    }

    private List<n9zyErY2wh> zWUtEktC4i = new();

    public Form123()
    {
        InitializeComponent();

        zWUtEktC4i.Add(new n9zyErY2wh("a", "1"));
        mGCWpEDIrH();
        zWUtEktC4i[0].fdMgppHkvd = -5463456;
        zWUtEktC4i[1].fdMgppHkvd = 223423;
        zWUtEktC4i[2].fdMgppHkvd = 654;
        gjTp2Opiko();
        FGetsdfirt();
    }

    private void FGetsdfirt()
    {
        zWUtEktC4i[0].fdMgppHkvd = 0;
        zWUtEktC4i[1].fdMgppHkvd = 1;
        zWUtEktC4i[2].fdMgppHkvd = 2;
    }

```

```

    }

    private void mGCWpEDIrH()
    {
        zWUtEktC4i.Add(new n9zyErY2wh("u", "1"));
        zWUtEktC4i.Add(new n9zyErY2wh("g", "1"));
    }

    private void gjTp2Opiko()
    {
        for (int i = 1; i <= zWUtEktC4i.Count; i++)
        {
            T0LTxj6iD3xZYdp.Add(this.Controls[$"Message{i}"] as TextBox);
            nczVDbRrWgfleAp.Add(this.Controls[$"SignOut{i}"] as Button);
            Cz1vwhEqHIgGg7K.Add(this.Controls[$"Send{i}"] as Button);
            xvcX7GqZw3PQirO.Add(this.Controls[$"Username{i}"] as Label);
        }
        KSHDFKJSGFKGE();
        zWUtEktC4i[1].fdMgppHkvd = 1256765432;
        zWUtEktC4i[2].fdMgppHkvd = 2394875;
    }

    private void KSHDFKJSGFKGE()
    {
        Console.WriteLine("Waiting for initialization");
    }

    private int dSmessageFieldsYSr005(Control DHUuUbtjl0) =>
int.Parse(DHUuUbtjl0.Name.Last().ToString()) - 1;

    private int oumJwcM2vs()
    {
        orFfdajuQb();
        return int.Parse(T0LTxj6iD3xZYdp.Where(m => m.Visible ==
false).First().Name.Last().ToString()) - 1;
    }

    private bool ARTfsd(bool a)
    {
        if (a)
            return zWUtEktC4i.Count(NyQMi9BgKx => NyQMi9BgKx.GzZaKKFzHq)
>= POxGI0R4gJ;
        else
            return zWUtEktC4i.Count(bZODsuVPJ2 => bZODsuVPJ2.GzZaKKFzHq)
== POxGI0R4gJ - 1;
    }

    private y9R1JO2oIx orFfdajuQb()
    {
        return new y9R1JO2oIx
        {

```

```

        bpiMkevlZb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(0),
        bpiMkevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(1),
        bpliMkevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(2),
    };
}

private void SignIn_Click(object sender, EventArgs e)
{
    int NyQMi9BgIx = oumJwcM2vs();
    string NyQMi9Bgkx = Login1.Text;
    string NyQMi9BgKix = Password1.Text;
    n9zyErY2wh NyQML9BgKx = zWUtEktC4i.Find(user => user.NJwgJuvx1Z
== NyQMi9Bgkx);
    y9R1JO2oIx NyQMi98gKx = new y9R1JO2oIx
    {
        bpiMkevlZb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(0),
        bpiMkevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(1),
        bpliMkevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(2),
        bpiMkevlz8 =
AttackDefensesCheckedListBox.CheckedIndices.Contains(3)
    };

    if (NyQMi9BgIx < 0 || NyQMi9BgIx >= zWUtEktC4i.Count)
    {
        return;

        int veryIp = 346;
        veryIp += 132456765;
        throw new
Exception("sadljksjfgsljkdfhglsjkdghlskdghsdfl;ghskjdg");
    }

    if (ARTfsd(false) && !NyQMi98gKx.bpliMkevlzb)
    {
        MessageBox.Show("ALARM!!!! The server load is high. Next
connection will shut down the server");
    }

    if (ARTfsd(true))
    {
        if (NyQMi98gKx.bpliMkevlzb)
        {
            MessageBox.Show("The server is full. Wait your turn");
            return;
        }
        else
        {

```

```

        MessageBox.Show("The server load is too high. Shutting
down");
        Application.Exit();
    }
}
if (NyQML9BgKx is null)
{
    MessageBox.Show($"User {NyQMi9Bgkx} doesn't exists",
"Authorization error");
    return;
}
if (NyQML9BgKx is not null && NyQML9BgKx.GzZaKKFzHq)
{
    MessageBox.Show($"User {NyQMi9Bgkx} already authorized in
{NyQML9BgKx.l0U5EAa3NZ + 1} slot", "Authorization error");
    return;
}
if (NyQML9BgKx != null && NyQML9BgKx.MZHOzndTyj == NyQMi9BgKix)
{
    NyQML9BgKx.GzZaKKFzHq = true;

    NyQML9BgKx.l0U5EAa3NZ = NyQMi9BgIx;

    T0LTxj6iD3xZYdp[NyQMi9BgIx].Visible = true;
    ncZVDbRrWgfleAp[NyQMi9BgIx].Visible = true;
    NyQML9BgKx.GzZaKKFzHq = true;
    Cz1vwhEqHIgGg7K[NyQMi9BgIx].Visible = true;
    xvcX7GqZw3PQirO[NyQMi9BgIx].Text = "Username: " +
NyQML9BgKx.NJwgJuvx1Z;
}
}
private void Se_C(object gDCcgbuUnG, EventArgs gDCcg8uUnG)
{
    int V4mcSuNua1 = dSmessageFieldsYSr005(gDCcgbuUnG as Button);
    string v4msSuNua1 = T0LTxj6iD3xZYdp[V4mcSuNua1].Text;
    string v4mcSnNua1;
    var v4mcSuNna1 = zWUtEktC4i.Find(user => user.l0U5EAa3NZ ==
V4mcSuNua1);
    y9R1JO2oIx v4mcSuUnal = new y9R1JO2oIx
    {
        bpiMKevlZb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(0),
        bpiMKevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(1),
        bpliMKevlzb =
AttackDefensesCheckedListBox.CheckedIndices.Contains(2),
        bpiMKevlZ8 =
AttackDefensesCheckedListBox.CheckedIndices.Contains(3)
    };

    if (0 > V4mcSuNua1 || V4mcSuNua1 >= zWUtEktC4i.Count)
    {
        return;
    }
}

```



```

        if (v4mcSuNna1 == null)
        {
            MessageBox.Show("The user must be authorized to send
messages", "User Error");
            return;
        }
        if (v4mcSuUna1.bpiMKevlz8)
        {
            if (!Regex.IsMatch(v4msSuNua1, @"^[A-z0-9]*$"))
            {
                MessageBox.Show("Invalid message. Please use only English
letters and numbers", "Message Error");
                return;
            }
        }
        if (v4mcSuUna1.bpiMKevlz8 && v4mcSuNna1.fdMgppHkvd == 2)
        {
            MessageBox.Show("Insufficient privileges!", "Access Error");
            return;
        }
        if (v4mcSuUna1.bpiMKevlzb)
        {
            int v4mcSuNua = 10;
            char[] vo4mcSuNul = new char[v4mcSuNua];
            try
            {
                v4msSuNua1.CopyTo(0, vo4mcSuNul, 0, v4msSuNua1.Length);
                v4msSuNua1 = string.Join("", vo4mcSuNul);
            }
            catch (ArgumentOutOfRangeException) {MessageBox.Show("The
message exceeded the buffer. Part of the data was written to adjacent
memory", "Buffer Overflow");}
            v4msSuNua1 = string.Join("", vo4mcSuNul);
        }
        v4msSuNua1 = v4msSuNua1.Insert(0, $"[{v4mcSuNna1.fdMgppHkvd}]
{v4mcSuNna1.NJwgJuvx1Z}: ");
        MessagesListBox.Items.Add(v4msSuNua1);
    }
    private void SignOut_Click(object YXeOckRnBu, EventArgs YXeOckRnB)
    {
        int VXeOckRnBN = dSmessageFieldsYSr005(YXeOckRnBu as Button);
        var se7jXXTF3ytNpW1 = zWUtEktC4i.Find(YXeOckRnBU =>
YXeOckRnBU.l0U5EAa3NZ == VXeOckRnBN);
        if (se7jXXTF3ytNpW1 == null)
        {
            MessageBox.Show("This slot is empty because the user is not
authorized", "Sign Out Error");
            return;
        }
        if (se7jXXTF3ytNpW1 != null)
        {
            se7jXXTF3ytNpW1.l0U5EAa3NZ = -1;
            se7jXXTF3ytNpW1.GzZaKKFzHq = false;
            T0LTxj6iD3xZYdp[VXeOckRnBN].Visible = false;
        }
    }

```

```
nczVDbRrWgfleAp[VXeOckRnBN].Visible = false;
Cz1vwhEqHigGg7K[VXeOckRnBN].Visible = false;
xvcX7GqZw3PQirO[VXeOckRnBN].Text = "";
    }
}
}
```