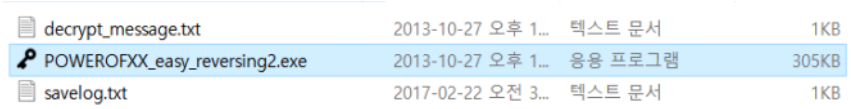
Easy2.Net

2017년 2월 22일 수요일 오전 3:01

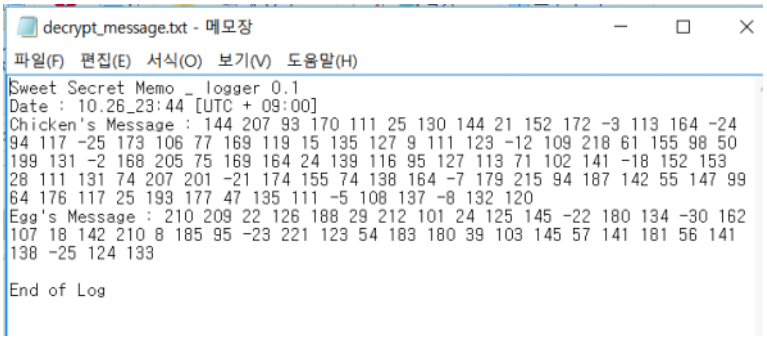
2013 Power Of XX Qual 문제이다. 출제자는 성원이형이다.

바이너리를 실행시키면 다음과 같다.

Plain Message에 원하는 문자열을 적어 넣고 submit을 넣으면 Crypted Message가 나온다.
이때 Crypted Message는 파일 형식으로 저장된다.



앗 위의 decrypt_message.txt는 뭘까?!



바로 출제자가 적어 놓은 문자열이다.
이 문자열을 복호화 하는 문제 인 것 같다.

우선 이 프로그램을 ilspy에 넣어 보았다.

```
using ...

namespace POWEROFXX_easy_reversing
{
    public class Form1 : Form
    {
        public string d8jergu394r0nnsjd94jfs = "9pMaVs5DxiOPGe8JETXYmg3lbudro6Qk1WLKwyhfnS4Iv0ABtjUCc7RZz2NFHq";

        public int njgcgcxdxxx6r = DateTime.Today.DayOfYear;

        public int zfgvjnkji8y6ug9u9i = DateTime.Now.Hour;

        public int cljbyt798ygdre5 = DateTime.Now.Minute;

        public int zsawsrf6g0i98t6vllp = DateTime.Today.Month;

        public int qexyg8j9u8thuhg = DateTime.Today.Day;

        private IContainer components;

        private Button button1;

        private Panel panel1;

        private Panel panel2;

        private Label label3;

        private TextBox textBox1;

        private TextBox textBox2;

        private Label label1;

        private void textbox1_mouseclick(object sender, MouseEventArgs e)
        {
        }

        public Form1()
        {
            this.InitializeComponent();
            this.d8jergu394r0nnsjd94jfs += "KfeROdEILJs5W6D1m4XFtH7YbwgrUConPuqQBcSxT092zljv8yMAGhpZN3akVi";
            this.d8jergu394r0nnsjd94jfs += "8vxekVPpYlsXDAujWoJEingTGf3mCh59LROt6cdUNMb41zH7Kr0yS2BIFZqawQ";
        }
```

다음과 같이 코드가 나오는데, 의도적으로 난독화를 해 놓은 것을 볼 수 있다. 따라서 나는 sublime text에 코드를 붙여넣고, 난독화 된 변수나 함수는 의미를 알아낸 후 이름을 바꿔가며 풀었다.

먼저 프로그램의 흐름을 읽어보자.
처음에 main에서 Form1을 불러온다.

```
public Form1()
{
    this.InitializeComponent();
    this.StrangeString += "KfeROdEILJs5W6D1m4XFtH7YbwgrUConPuqQBcSxT092zljv8yMAGhpZN3akVi";
    this.StrangeString += "8vxekVPpYlsXDAujWoJEingTGf3mCh59LROt6cdUNMb41zH7Kr0yS2BIFZqawQ";
}
```

이때 InitializeComponent(); 를 부르는데, 이때 컴포넌트들을 셋팅한다.
이 곳을 보면 submit button을 찾을 수 있고, 이 버튼의 이벤트 핸들러까지 찾을 수 있다.

이벤트 핸들러는 다음과 같다.

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.textBox2.Text.Length > this.StrangeString.Length - this.Month)
    {
        MessageBox.Show("Length Error", "ERROR_bb");
        return;
    }
    this.gettime();
    string text = this.encodeStart(this.textBox2.Text);
    this.textBox1.Text = text;
    this.lfkfidngigiwhiu3yr89igorg(text);
}
```

이 이벤트 핸들러 함수에서 입력받은 문자열에 대해 암호화를 진행한 후 암호화된 문자열을 화면에 뿌려준다.
암호화는 다음 함수에서 실행된다.

```
public string encodeStart(string str)
{
    StringBuilder stringBuilder = new StringBuilder();
    string text = this.xorString(Form1.upsideDownInput(str));
    string[] array = text.Split(new char[]
    {
        ' '
    });
    for (int i = 0; i < array.Length - 1; i++)
    {
        stringBuilder.Append(this.changeNum(array[i], i) + " ");
    }
    return stringBuilder.ToString();
}
```

암호화 과정을 분석하고, 복호화 코드를 짜 보자.

이 함수에서 호출하는 중요한 함수가 3가지 있다.

1. upsideDownInput

이 함수는 다음과 같다.

```
public static string upsideDownInput(string input)
{
    int length = input.Length;
    char[] array = new char[length];
    for (int i = 0; i < input.Length; i++)
    {
        array[i] = input[length - i - 1];
    }
    return new string(array);
}
```

내가 바꿔준 이름 그대로 문자열을 뒤집어 주는 것이다. 이 함수에 대한 복호화 코드는 다음과 같이 짤 수 있다.

```
for(i=sizeof(chicken)/4-1;i>=0;i--)
{
    printf("%c",chicken[i]);
}
```

그냥 뒤집어서 출력해 주는 것이다.

2. xorString

이 함수는 다음과 같다.

```
public string xorString(string a)
{
    StringBuilder stringBuilder = new StringBuilder();
    int num = this.Month;           //10
    char[] array = a.ToCharArray(); //string a to char array
    for (int i = 0; i < a.Length; i++)
    {
        stringBuilder.Append((int)(this.StrangeString[i + num] ^ array[i]) + " ");  //StrangeString[i+10]^array[i]
    }
    return stringBuilder.ToString();
}
```

이 프로그램에 무작위의 문자열이 담겨있는 변수인 StrangeString이 있는데, 이 문자열 중 한 곳과 xor을 수행하는 부분이다. 다음과 같이 복호화 코드를 짤 수 있다.

```
for(i=0;i<sizeof(chicken)/4;i++)
{
    chicken[i]^=StrangeString[i+10];
}
```

xor의 특성상 xor해 준 대상과 그대로 xor 하면 원본 문자를 얻을 수 있다.

3. changeNum

이 함수는 다음과 같다.

```
public int changeNum(string chr, int range)
{
    int num = int.Parse(chr);   //num is char
    int num2 = range % 3;
    int num3 = 2;
    if (num2 == 0)
    {
        num += this.Day * num3 + this.Minute * num3 - this.Hour * 2;     //num = num + 26 * 2 + 44 * 2 - 23 * 2
                                                                         //num = num + 94
    }
    else if (num2 == 1)
    {
        num += this.Month * 3 + this.Minute * 2 - this.Hour * num2;      //num = num + 10 * 3 + 44 * 2 - 23 * 1
                                                                         //num = num + 95
    }
    else if (num2 == 2)
    {
        //num = num + 299 - 10 * 10 - 44 * 2 - 23 * 6 - 4
        //num = num - 31
        num += this.DayOfYear - this.Month * (num2 * 5) - this.Minute * num2 - this.Hour * (num3 + 4) - num2 * num3;
    }
    return num;
}
```

입력한 range를 3으로 %연산자를 수행한 후, 나온 결과값에 따라 다른 연산을 수행한다. 이때 연산에 현재 날짜, 월 등이 사용되는데, 우리는 출제자가 암호화를 수행했을 당시의 시간을 사용해야 하는데, 이 시간은 텍스트 파일에 그대로 나와 있다. 복호화 코드를 다음과 같이 짤 수 있다.

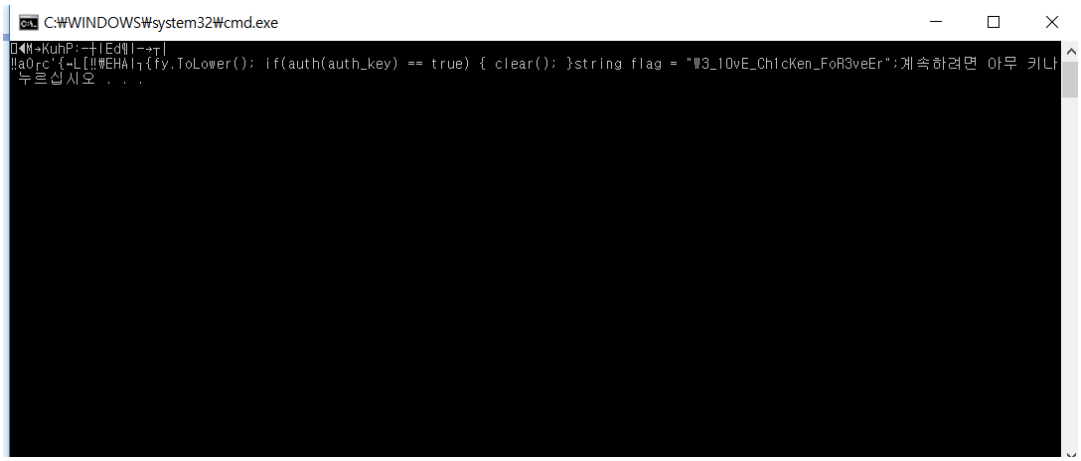
```
for(i=0;i<sizeof(chicken)/4;i++)
{
    if(i%3==0)
        chicken[i]-=94;
    else if(i%3==1)
        chicken[i]-=95;
    else if(i%3==2)
        chicken[i]+=31;
} //changeNum complete
```

다음과 같이 수행했다.

코드는 다음과 같다.

```
#include <stdio.h>
int main(void)
{
    int i;

    char
    StrangeString[]="9pMaVs5DxiOPGe8JETXYmg3lbudro6Qk1WLKwyhfnS4lv0ABtjUC
    c7RZz2NFHqHv8VzYa5b1FMGNODW4kwX9L3hK6SqsTtyxoE0Z7fPJlgrCAQiljBuenR
    cp2dUmfgnCw4HPJRdXKlq31YNDZMS82OjA7eUxpozasVmykiQrTFLW6htGb9B0lEc
    vu5jxLaZdWYngAfKGNhzTcXQU7Jy9sFbp0eRl1ECrv23PSw846oH5MBVtlDiOqumk
    U0tnl9bVK4iB2LzZXy7PaCHcAl5pOsSfjgqkr1vuRTFEo8Dxmhw3QGdeJM6WYNgWY
    N9w4LuPjxJl1MhOkniQy8CBUXr6THaKDctEdb0Imp32VfFZGvAS5ezqsR7owj3J9fL8
    QY2kArXKgOEzmSdqHpcMsn1ahGWxCe7yPllTuDRb6F40oZtiUBvV5NKfeROdElLJs
    5W6D1m4XFtH7YbwgrUConPuqQBcSxT092zljv8yMAGhpZN3akVi8vxekVPpYlsXDA
    ujWoJEingTGf3mCh59LROt6cdUNMb41zH7Kr0yS2BlFZqawQ";
    int
    chicken[]={144,207,93,170,111,25,130,144,21,152,172,-3,113,164,-24,94,117,-
    25,173,106,77,169,119,15,135,127,9,111,123,-12,109,218,61,155,98,50,199,131
    ,-2,168,205,75,169,164,24,139,116,95,127,113,71,102,141,-18,152,153,28,111,
    131,74,207,201,-21,174,155,74,138,164,-7,179,215,94,187,142,55,147,99,64,17
    6,117,25,193,177,47,135,111,-5,108,137,-8,132,120};
    int
```

```
	egg[]={210,209,22,126,188,29,212,101,24,125,145,-22,180,134,-30,162,107,18
	,142,210,8,185,95,-23,221,123,54,183,180,39,103,145,57,141,181,56,141,138,-
	25,124,133};

	for(i=0;i<sizeof(chicken)/4;i++)
	{
		if(i%3==0)
			chicken[i]-=94;
		else if(i%3==1)
			chicken[i]-=95;
		else if(i%3==2)
			chicken[i]+=31;
	} //changeNum complete

	for(i=0;i<sizeof(chicken)/4;i++)
	{
		chicken[i]^=StrangeString[i+10];
	}

	for(i=sizeof(chicken)/4-1;i>=0;i--)
	{
		printf("%c",chicken[i]);
	}

	for(i=0;i<sizeof(egg)/4;i++)
	{
		if(i%3==0)
			egg[i]-=94;
		else if(i%3==1)
			egg[i]-=95;
		else if(i%3==2)
			egg[i]+=31;
	} //changeNum complete

	for(i=0;i<sizeof(egg)/4;i++)
	{
		egg[i]^=StrangeString[i+10];
	}

	for(i=sizeof(egg)/4-1;i>=0;i--)
	{
		printf("%c",egg[i]);
	}
	return 0;
}
```



성공!