

Easy.Net(아직 덜 품) 윤희관

Easy.Net을 IISpy로 열어보았습니다.

```
private int[] jkd9j349njw984nt0wner9g34j = new int[]
{
    7,
    6,
    5,
    8,
    9,
    4,
    2,
    3,
    1,
    0
};

public string odf03mfg03mn4__dsfij9834rosdf043 = "MXgLk4UD5zE0jTuYy5/7MaQzMNQuAcJu6yEiLzi30Rb0ISlhr7orIyY40tTWw7+TdAIPqsY+oL/";
```

맨 첫 부분에 이러이러한 변수들이 나옵니다.

밑에서 변수들로 이것저것 하는 것을 보니 필시 필요할 것이므로 적어 두었습니다.

밑으로 내리니

```
this.InitializeComponent();
this.label1.BackColor = Color.Transparent;
this.jkd9j349njw984nt0wner9g34j[this.jkd9j349njw984nt0wner9g34j.Length - this.jkd9j349njw984nt0wner9g34j.Length] = 1;
this.odf03mfg03mn4__dsfij9834rosdf043 += "iV07R3sopmA+Z/CmoNdKAHIjwEJK+wc1V8ONG083f0S1BfWwEYbaoVXru5zhBgy83HSPPNjb0IJOUsUPmSC66ByUd";
this.odf03mfg03mn4__dsfij9834rosdf043 += "CJZMwRLg72IsofYYEV/I2231K2xqh11HFRz0qaZPp";
this.odf03mfg03mn4__dsfij9834rosdf043 += "C/FyWdhkRAXEccfn5v513iz+Qs7WI=";
```

아까 위에 저 부분에 추가되는 String이 있습니다. =로 끝나는 것을 보아 base64의 냄새가 나기 때문에 다짜고짜 돌려보았습니다! (생각나는게 그것 밖에 없었습니다.)

..

MXgLk4UD5zEOjTuYy5/7MaQzMNquAcJu6yEiLzi30Rb0ISlhr7orlyY4OtTWW7+TdAl
PqsY+oL/iVO7R3sopmA+Z/CmoNdKAHljwEJK+wclV8ONGO83f0SIBfWwEYbaoVXr
u5zhBgy83HSPpNjb0lJOUsUPmSC66ByUdCJZMwRLg72IsofYYEV/I223IK2xqh11
HFRzOqaZPpC/FyWDhkRAXEccfn5v513iz+Qs7WI=

< DECODE >

UTF-8



(You may also select input charset.)

1x3L1B,*130r4n"/8T)a+&8:[t1>T)5ÇL+UF;)A)l^aUz8A/7<#66 \$
Ll"Y3J`K`E0#m+uLTs:>| %D@\\GL~o],b

(아무 일도 일어나지 않았다...)

마음을 가다듬고 버튼을 누르면 key가 넘어가는 것이므로 Button에 관련된 함수
들을 보기로 했습니다. 함수 선언 되있는 거 다 빼고 button 들에 대해 확인해보
기 위해 InitializeComponent()를 보니

```
this.button3.TabIndex = 17;  
this.button3.Text = "y3kt3g";  
this.button3.UseVisualStyleDef
```

button1은 submit, button2와 4,5는 Z2V0a2V5 인데 애만 다릅니다..

수상한 녀석은 탐험해줘야 합니다.

```
private void button3_Click(object sender, EventArgs e)  
{  
    byte b = 49;  
    string[] array = this.dfj4rfioksdfiop4ewjf9dsr34r();  
    object arg_1E_0 = 2;  
    object arg_1E_1 = array[14];  
    char c = (char)b;  
    string str = arg_1E_0 + arg_1E_1 + c.ToString();  
    str += this.dfj4rfioksdfiop4ewjf9dsr34r()[1];  
    byte b2 = 95;  
    char c2 = (char)b2;  
    string text = c2.ToString() + "!" + this.jkd9j349njw984nt0wner9g34j[this.jkd9j349njw984nt0wner9g34j.Length - 1];  
    text += this.jf9j39fjd9rjngsweetfub95f();  
    this.label2.Text = str + text;  
    int arg_8C_0 = this.jkd9j349njw984nt0wner9g34j[6];  
}
```

제가 아까 모아놔던 수상한 녀석들이 나타났습니다. dfj4r~~은 밑 부분에서 찾은
친구인데 저 부분에 있는 연산대로 해봅니다

뭔가 여기서 키 값을 찾는 것 같으니 계산해봅시다.

b는 49.. array에는 3EIGA4BEP06Ga7o5가 들어가고 길이가 16

arg_1E_0 = 2, 1E_1 = array[14] = o, char c = (char)b 인 걸 보아 아스키

스트링에 3... 등등 다 모아서 정리해보았더니

str = 2o1E가 나온다. text를 계산하던 중 jf9j39~라는 걸 호출하는데 기존에 jkd9~~를 수상하게 작업합니다.

```
private void button2_Click(object sender, EventArgs e)
{
}

public string jf9j39fjd9rjngsweetfub95f()
{
    this.dfj4rfioksdfiop4ewjf9dsr34r();
    int num = this.jkd9j349njw984nt0wner9g34j[2];
    string arg = "-" + this.dfj4rfioksdfiop4ewjf9dsr34r()[14];
    string result = arg + num;
    for (int i = 0; i < this.jkd9j349njw984nt0wner9g34j.Length; i++)
    {
        this.label2.ForeColor = Color.FromArgb(15, 15, 15);
        num >>= this.jkd9j349njw984nt0wner9g34j[i];
        num += this.jkd9j349njw984nt0wner9g34j[i];
        num <<= this.jkd9j349njw984nt0wner9g34j[i];
    }
    return result;
}
```

jf9j39를 계산해보면 result의 num 부분까진 스무스하게 할 수 있습니다.

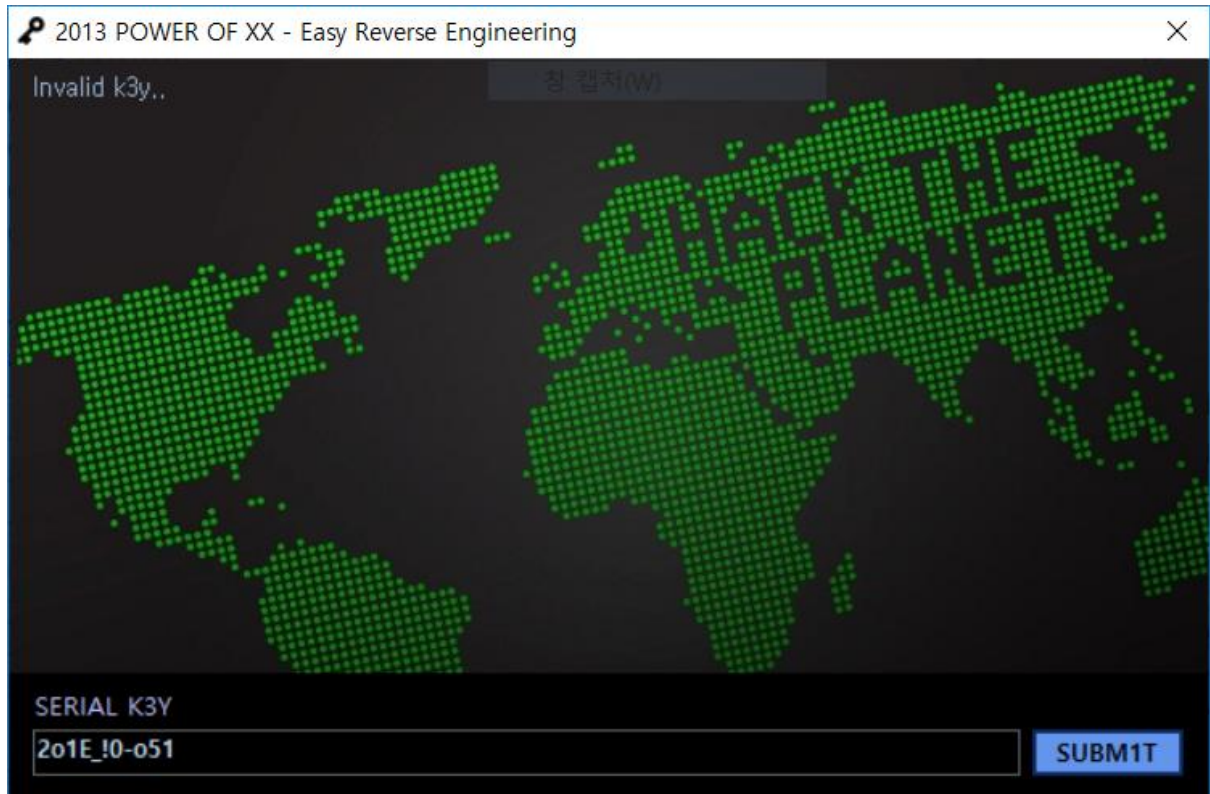
result = arg + num 은 -o5 이고 밑에 부분은 어떠한 작업을 수행하여 줍니다..

c언어로 확인해보면 num은 1538입니다.

button3로 돌아와보니 쓸 때 없는 것 같습니다 ㅋㅋㅋ 여튼..

text = _!0-o5가 됩니다.

arg에 모든 연산을 끝내면 2o1E_!0-o51이 됩니다. 집어 넣어 봅시다.



아무 일도 일어나지 않습니다.. 2013년 POX여서 2o1E까지는 뭔가 그럴싸 한거 같
같은 뒷 부분이 의미 없어 보이니 좀 더 분석을 해봐야 될 것 같습니다.!