[2013 POWER OF XX - easy reverse engineering]

주어진 프로그램을 실행하면 아래와 같은 모습이 나온다.

시리얼 키를 입력하라고 한다. 좌측 상단에서 HACK THE PALNET (오타?) 을 클릭하라고 하므로 지도에 있는 글씨를 클릭해 보았다.

base64로 암호화된 문자열이 나온다. 복호화해 보면 http://ilspy.net/ 이라는 주소가 나온다. 들어가 보면 디컴파일러를 다운받을 수 있다.

ILSpy로 디컴파일해서 분석을 해보자

ILSpy로 열어서 디컴파일하고 코드를 보다보면 암호화 부분이 보인다.

```
public string tpyrcde821SEA(string Input, string key)
{
    string result;
    try
    {
        RijndaelManaged rijndaelManaged = new RijndaelManaged();
        byte[] array = Convert.FromBase64String(Input);
        byte[] bytes = Encoding.ASCII.GetBytes(key.Length.ToString());
        PasswordDeriveBytes passwordDeriveBytes = new PasswordDeriveBytes(key, bytes);
        ICryptoTransform transform = rijndaelManaged.CreateDecryptor(passwordDeriveBytes.GetBytes(32), passwordDeriveBytes.GetBytes(16));
        MemoryStream memoryStream = new MemoryStream(array);
        CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Read);
        byte[] array2 = new byte[array.Length];
        int count = cryptoStream.Read(array2, 0, array2.Length);
        memoryStream.Close();
        cryptoStream.Close();
        string @string = Encoding.Unicode.GetString(array2, 0, count);
        result = @string;
    }
    catch
    {
        result = "Invalid k3y..";
    }
    return result;
}

public string tpyrcen821SEA(string Input, string key)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    byte[] bytes = Encoding.Unicode.GetBytes(Input);
    byte[] bytes2 = Encoding.ASCII.GetBytes(key.Length.ToString());
    PasswordDeriveBytes passwordDeriveBytes = new PasswordDeriveBytes(key, bytes2);
    ICryptoTransform transform = rijndaelManaged.CreateEncryptor(passwordDeriveBytes.GetBytes(32), passwordDeriveBytes.GetBytes(16));
    MemoryStream memoryStream = new MemoryStream();
    CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
    cryptoStream.Write(bytes, 0, bytes.Length);
    cryptoStream.FlushFinalBlock();
    byte[] inArray = memoryStream.ToArray();
    memoryStream.Close();
    cryptoStream.Close();
    return Convert.ToBase64String(inArray);
}
```

C#은 처음이라 많이 당황스럽다. RijndaelManaged를 구글에 찾아보니 래퍼런스가 나온다. 무슨 말인지는 잘 모르겠지만 한 가지 확실한건 대칭키를 이용한 암호화를 한다는 것이다. 따라서 코드를 보다보면 대칭키가 숨겨져 있을 것이다.

찾아보면 암호화 하는 함수를 호출하는 부분은 없는 것 같지만 복호화 하는 함수를 호출하는 부분은 있다.

```
private void button1_Click_1(object sender, EventArgs e)
{
    string.Concat(new object[]
    {
        this.dfj4rfioksdfiop4ewjf9dsr34r()[0],
        this.nnumbobeeerofowoooeneneee,
        this.dfj4rfioksdfiop4ewjf9dsr34r()[4],
        this.dfj4rfioksdfiop4ewjf9dsr34r()[this.dfj4rfioksdfiop4ewjf9dsr34r().Length - 1]
    });
    this.label1.Text = this.tpyrcde821SEA(this.odf03mfg03mn4__dsfij9834rosdf043, this.textBox2.Text);
}
```

우리가 입력한 값을 key로 넣고 암호화 되어있는 문자열을 input으로 넣는다. 대칭키를 찾아서 입력하면 암호화 되어있는 분자열이 해독되는 것 같다.

문제를 풀기 위해 대칭키를 찾아야 하므로 코드를 더 살펴봤다. 이것저것 보면서 삽질을 하다가 문자열을 만드는 듯한 함수가 있길래 한번 살펴봤다.

```
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];
}
```

복잡해 보이지만 한줄 한줄 읽으면서 따라가면 아주 단순한 녀석이다. 이 함수가 진행되면서 만들어지는 문자열은 '2o1E_!0-o5' 이다. 뭔가 답인 것 같아서 넣어봤더니 flag가 떴다.;;

풀기는 풀었는데 코드가 어떻게 동작하는지 이해가 안돼서 코드를 조금 살펴봤다.

```
this.button3.Text = "y3kt3g";
```

“g3tk3y“를 거꾸로 읽으면 "g3tk3y"....
이걸 일찍 봤으면 엄청 빨리 풀었을 것 같다.

또 찾아보니
this.Button1.Click += new System.EventHandler(this.myEventHandler);
this.Button2.Click += new System.EventHandler(this.myEventHandler);
과 같은 형식으로 버튼을 연결(?) 할 수 있다고 한다. 그래서 버튼 한 개로 button1,2,3을 모두 동작 시킨 것 같다.