Codeengn Challenges Advance RCE LEVEL7 풀이

Reverse2 L07 Start

Author : HMX0101

Korea :

Name이 CodeEngn일때 Serial은 28BF522F-A5BE61D1-XXXXXXXX 이다.

XXXXXXXX 를 구하시오

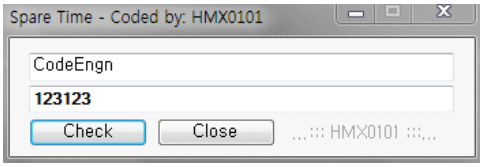
English :

When the Name is CodeEngn, the Serial is 28BF522F-A5BE61D1-XXXXXXXX.

Find XXXXXXXX

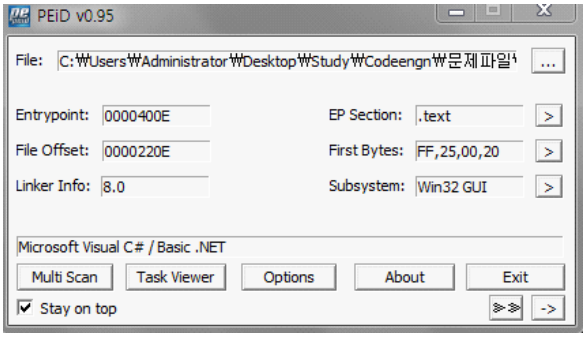
Down

프로그램을 실행시켜보니 이름과 시리얼을 입력해 맞추는 프로그램 이었다.



그런데 아무리 check를 눌러도 아무런 반응도 없었다.

PEID로 확인해보니 .NET 프로그램이었다.



.Net으로 컴파일된 프로그램은 .NET reflector 프로그램을 이용해 디컴파일해서 분석을 하는게 좋을것 같아서 .NET reflector로 디컴파일을 하였다.

버튼을 클릭할때 나는 이벤트코드를 찾아보았다.

```
private void button1_Click(object sender, EventArgs e)
{
    string str = "";
    string str2 = "";
    string str3 = "";
    ytrewq ytrewq = new ytrewq();
    if ((((this.textBox1.Text.Length >= 5) && (this.textBox1.Text.Length <= 0x1b)) && ((this.textBox2.Text.Length == 0x1a) && (this.textBox2.Text[8] =
    {
        for (int i = 0; i < 8; i++)
        {
            str = str + this.textBox2.Text[i];
        }
        uint num = Convert.ToUInt32(str, 0x10);
        for (int j = 9; j < 0x11; j++)
        {
            str2 = str2 + this.textBox2.Text[j];
        }
        uint num2 = Convert.ToUInt32(str2, 0x10);
        for (int k = 0x12; k < 0x1a; k++)
        {
            str3 = str3 + this.textBox2.Text[k];
        }
        uint fsfsdf = Convert.ToUInt32(str3, 0x10);
        uint num5 = ytrewq.qwerty(dfgsf(this.textBox1.Text));
        uint hashCode = (uint) this.textBox1.Text.GetHashCode();
        fsfsdf ^= hashCode;
        this.yreee[0] = num;
        this.yreee[1] = num2;
        this.yreee[2] = num;
        this.yreee[3] = num2;
        if ((this.vxzzz(this.yreee, this.ewrrr, 0x8ffe2225, fsfsdf) && (this.yreee[2] == hashCode)) && (this.yreee[3] == num5))
        {
            MessageBox.Show("Congratulations, mate!", "Fine!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        }
    }
}
```

다음과 같은 코드가 나오는데 중요한 것들을 분석해보겠다.

```
if ((((this.textBox1.Text.Length >= 5) && (this.textBox1.Text.Length <= 0x1b)) && ((
```

이 if문은 tesxtBox에 있는 내용들을 검열하는건데 , 이름을 입력하는 text박스의 길이가 0x5~0x1b가 되지않으면 아래 문장을 실행시키지 않는다.

```
((this.textBox2.Text.Length == 0x1a) && (this.textBox2.Text[8] == '-'))) && (this.textBox2.Text[0x11] == '-'))
```

그리고 시리얼의 길이는 0x1a가 되어야 하고 8,11인덱스에 "-"이 있어줘야 한다.

```
for (int i = 0; i < 8; i++)
{
    str = str + this.textBox2.Text[i];
}
uint num = Convert.ToUInt32(str, 0x10);
for (int j = 9; j < 0x11; j++)
{
    str2 = str2 + this.textBox2.Text[j];
}
uint num2 = Convert.ToUInt32(str2, 0x10);
for (int k = 0x12; k < 0x1a; k++)
{
    str3 = str3 + this.textBox2.Text[k];
}
uint fsfsdf = Convert.ToUInt32(str3, 0x10);
```

그리고 각 하이픈 사이의 문자를 정수화 시킨다.(여기서 CodeEngn에 예에 나와있는것처럼 XXXXXXX같이 16진수 범위에 벗어나면 에러가뜨므로 조심해야 한다.)

```
uint num5 = ytrewq.qwerty(dfgsf(this.textBox1.Text));
uint hashCode = (uint) this.textBox1.Text.GetHashCode();
```

HashCode와 num5를 구한다.

```
fsfsdf ^= hashCode;
this.yreee[0] = num;
this.yreee[1] = num2;
this.yreee[2] = num;
this.yreee[3] = num2;
```

Fsfdf를 자신과 hashcode로 xor해준후 num과 num2값을 배열에 저장해준다.

```
if ((this.vxzzz(this.yreee, this.ewrrr, 0x8ffe2225, fsfsdf) && (this.yreee[2] == hashCode)) && (this.yreee[3] == num5))
{
    MessageBox.Show("Congratulations, mate!", "Fine!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
}
```

그 후 배열 두개와 0x8ffe2225, fsfdf를 어떤 함수로 전달해주고 저 함수를 시행 후의 배열과 hashcode, num5를 비교해서 밑의 문장을 실행해 주고있다.

저곳에서 호출해주는 함수를 분석해야 답을 얻을 수 있는거 같아서 저 함수를 분석해 보기로했다.

```
private bool vxzzz(uint[] rwerqw, uint[] kgtsdfs, uint pgdsfa, uint fsfsdf)
{
    pgdsfa ^= fsfsdf;
    uint num = (pgdsfa % 0x39) - 1;
    uint num10 = rwerqw[0];
    uint num11 = rwerqw[1];
    uint num6 = num;
    uint num2 = pgdsfa << ((byte) (0x61 ^ (num + 0x44)));
    if (num != 0)
    {
        while (num-- > 0)
        {
            uint num5 = num6 / 0x10;
            uint num3 = num10 << ((byte) (num6 / 8));
            uint num4 = num10 >> (3 + ((byte) num5));
            uint num7 = (num6 / 4) + 3;
            uint num9 = num7;
            num7 = kgtsdfs[(int) ((IntPtr) ((num2 >> ((byte) num7)) % 4))];
            uint num8 = num2 + num7;
            num11 -= (((num3 ^ num4) + num10) ^ num8) - num;
            num2 -= pgdsfa;
            num11 -= num;
            num3 = num11 << (((byte) (num9 + 1)) ^ 8);
            num4 = num11 >> (((byte) (((num6 / 2) - num9) + 0x17)) ^ 0x19);
            if (num == num6)
            {
                num11 ^= num;
            }
            if (num == ((num6 / 2) + (num9 ^ 0x1b)))
            {
                num9 = (num3 ^ num4) + (num11 ^ num);
            }
            else
            {
                num9 = (num3 ^ num4) + num11;
            }
            num10 -= num9 ^ (num2 + kgtsdfs[(int) ((IntPtr) (num2 & 3))]);
        }
        rwerqw[0] = num10 ^ 4;
        rwerqw[1] = num11 ^ 7;
        rwerqw[2] = rwerqw[1] ^ ((byte) (((num6 + 1) / 3) - 4));
        rwerqw[3] = rwerqw[0] ^ ((byte) (((num6 - 0x15) + 1) ^ 8));
        rwerqw[0] ^= kgtsdfs[4];
        rwerqw[1] ^= kgtsdfs[5];
        return true;
    }
    return false;
}
```

Pgdsfa와 fsfsdf를 xor해주고, 그값을 %39 -1 한 값이 0이 아니면 일정한 루프를 돌고 xor연산을 진행해 true를 반환하고, 0이면 false를 반환한다,
그리고 루프를 시작하면서 num값을 16,8,4로 나눠줘 하는데 여기서 num값이 16,8,4의 공배수인 16의 배수인것 같았다. 즉 16,32가 될 수 가 있다.
우리가 의도해야할 것은 이 함수의 결과로 true가 반환 되어야 하고, 또 이 함수로인해 배열에 저장된 결과가 hashcode와 num5와 값이 같아야 한다.

그러기 위해서는 XXXXXXXX부분을 어떻게든 초기에 알아내야하는데

이 문장을 이용해 XXXXXXXX값을 알아냈다.

```
private bool vxzzz(uint[] rwerqw, uint[] kgtsdfs, uint pgdsfa, uint fsfsdf)
{
    pgdsfa ^= fsfsdf;
    uint num = (pgdsfa % 0x39) - 1;
```

이 함수에 인자로 전달되는 pgdsfa는 0x8ffe2225다. 그리고 fsfsdf 는 fsfsdf ^ hashcode다 .

Messagebox.show 메서드를 이용해 메시지 창으로 hashcode를 알아본 결과 hashcode = 2652795861 = 9E1E73D5 였다.

Num의 최대값을 32라고 생각하면 % 39한 값은 33
Hashcode는 fsfsdf ^ 9E1E73D5 이니

0x8FFFE2225 ^ 0x9E1E73D5 ^ 33을 하니 답이 나왔다! :D