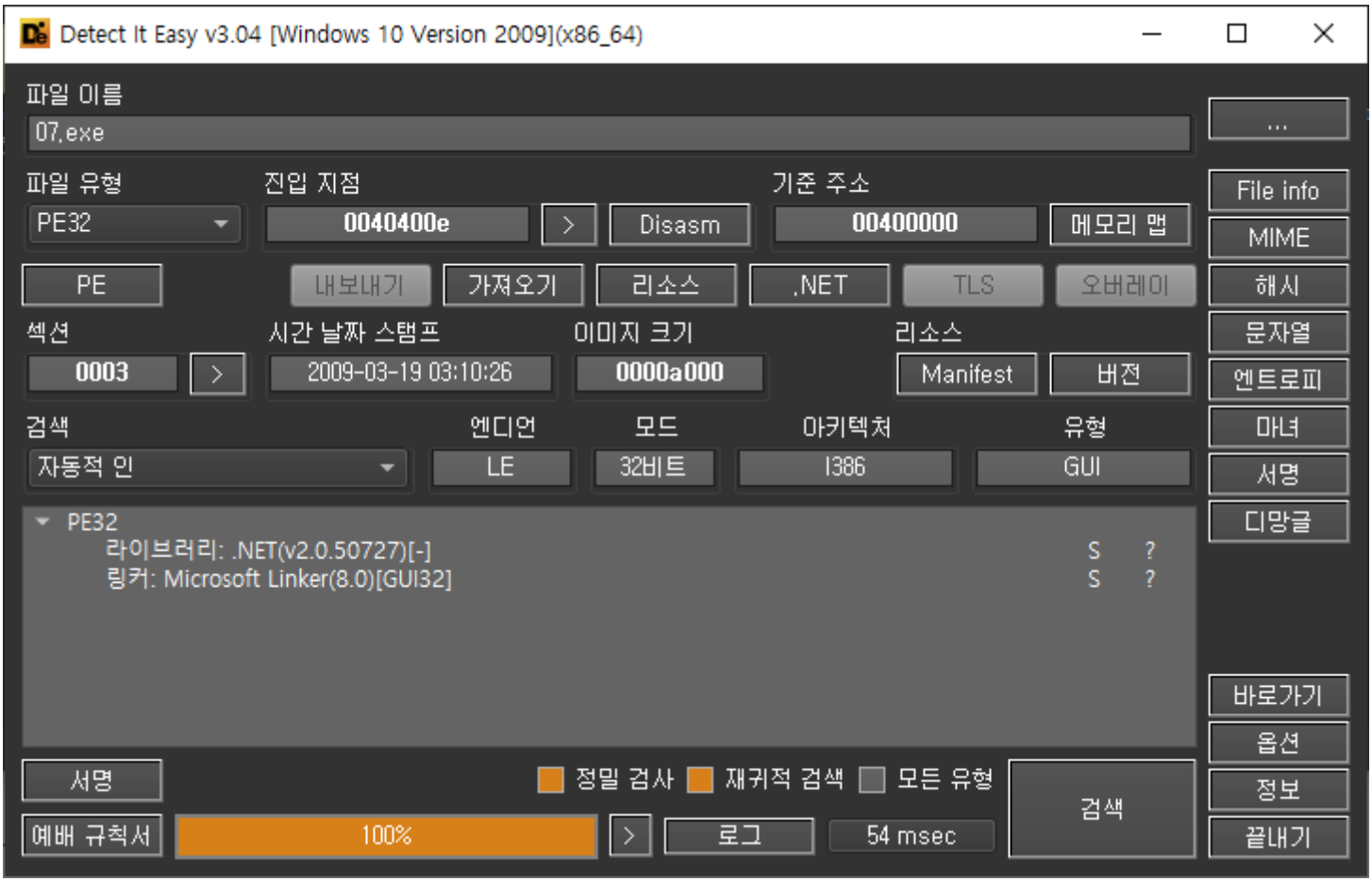
codeengn-advance-L07 풀이

리버싱 문제풀이 / WonIf / 2022. 5. 11. 21:40

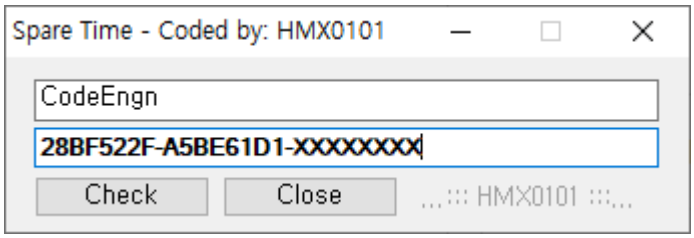
문제는 Name이 CodeEngn일 때 Serial을 원하고 있다.

Die로 열어본다.



.NET 프레임워크로 구현되어 있다. 이런 실행 파일은 Dnspy나 dotPeek으로 코드를 전부 볼 수 있다.

프로그램을 실행시켜본다.



문제에서는 이 아래의 시리얼을 원하는 것 같다.

dotPeek으로 열어본다.

```
    this.yreee[3] = uint32_2;
    if (!this.vxzzz(this.yreee, this.ewrrr, 2415796773U, fsfsdf) || (int) this.yreee[2] != (int) hashCode || (int) this.yreee[3] != (int) num1)
      return;
    int num2 = (int) MessageBox.Show("Congratulations, mate!", "Fine!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
  }
```

성공을 출력하는 구문을 찾았다.

if문의 조건 3가지**(this.vxzzz(this.yreee, this.ewrrr, 2415736773U, fsfsdf) / (int) this.yreee[2] == (int) hashCode / (int) this.yreee[3] == (int) num1)**를 만족하는 값을 찾아야 한다.

코드를 더 보게되면 시리얼에는 16진수만 와야 하고, 다른 값을 입력하면 예외 처리가 된다.
이제 코드 전체를 보지 않아도 브루트포스를 통해 쉽게 답을 찾을 수 있을 것 같다.

먼저, 코드를 수정하면서 검증해보기 위해 Dnspy로 열어본다.

```
145             string text = "";
146             string text2 = "";
147             string text3 = "";
148             ytrewq ytrewq = new ytrewq();
149             if (this.textBox1.Text.Length >= 5 && this.textBox1.Text.Length <= 27 && this.textBox2.Text.Length == 26 && this.textBox2.Text[8] == '-' && this.textBox2.Text[17] == '-')
150             {
151                 for (int i = 0; i < 8; i++)
152                 {
153                     text += this.textBox2.Text[i];
154                 }
155                 uint num = Convert.ToUInt32(text, 16);
156                 for (int j = 9; j < 17; j++)
157                 {
158                     text2 += this.textBox2.Text[j];
159                 }
160                 uint num2 = Convert.ToUInt32(text2, 16);
161                 for (int k = 18; k < 26; k++)
162                 {
163                     text3 += this.textBox2.Text[k];
164                 }
165                 uint num3 = Convert.ToUInt32(text3, 16);
166                 uint num4 = ytrewq.qwerty(Form1.dfgsf(this.textBox1.Text));
167                 uint hashCode = (uint)this.textBox1.Text.GetHashCode();
168                 num3 ^= hashCode;
169                 this.yreee[0] = num;
170                 this.yreee[1] = num2;
171                 this.yreee[2] = num;
172                 this.yreee[3] = num2;
173                 bool flag = this.vxzzz(this.yreee, this.ewrrr, 2415796773U, num3);
174                 if (flag && this.yreee[2] == hashCode && this.yreee[3] == num4)
175                 {
176                     MessageBox.Show("Congratulations, mate!", "Fine!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
177                 }
178             }
179         }
180
181         // Token: 0x06000008 RID: 11 RVA: 0x00002764 File Offset: 0x00000964
182         private void button2_Click(object sender, EventArgs e)
183         {
184             Application.Exit();
185         }
186
187         // Token: 0x0600000C RID: 12 RVA: 0x0000276C File Offset: 0x0000096C
188         private void Form1_Load(object sender, EventArgs e)
```

선택된 부분을

```
private void button1_Click(object sender, EventArgs e)
{
    string text = "";
    string text2 = "";
    string text3 = "";
    ytrewq ytrewq = new ytrewq();
    if (this.textBox1.Text.Length >= 5 && this.textBox1.Text.Length <= 27 && this.textBox2.Text.Length == 26 && this.textBox2.Text[8] == '-' && this.textBox2.Text[17] == '-')
    {
        for (int i = 0; i < 8; i++)
        {
            text += this.textBox2.Text[i].ToString();
        }
        uint num = Convert.ToUInt32(text, 16);
        for (int j = 9; j < 17; j++)
        {
            text2 += this.textBox2.Text[j].ToString();
        }
        uint num2 = Convert.ToUInt32(text2, 16);
        for (int k = 18; k < 26; k++)
        {
            text3 += this.textBox2.Text[k].ToString();
        }
        for (uint l = 0U; l <= 4294967295U; l += 1U)
        {
            uint num3 = l;
            uint num4 = ytrewq.qwerty(Form1.dfgsf(this.textBox1.Text));
            uint hashCode = (uint)this.textBox1.Text.GetHashCode();
            num3 ^= hashCode;
            this.yreee[0] = num;
            this.yreee[1] = num2;
            this.yreee[2] = num;
            this.yreee[3] = num2;
            if (this.vxzzz(this.yreee, this.ewrrr, 2415796773U, num3) && this.yreee[2] == hashCode && this.yreee[3] == num4)
            {
                MessageBox.Show("Congratulations, mate!", "Fine!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
                return;
            }
        }
    }
```

이런식으로 바꿔주어 0x0 ~ 0xffffffff까지 브루트 포스를 통해 메시지 박스를 띄워 보면,

정답을 알 수 있게 된다.

하지만 GetHashcode의 영향으로 32비트와 64비트의 값은 달랐고, 32비트 운영 체제를 설치하여 컴파일 하니 정답이 도출 되었다.

성공!