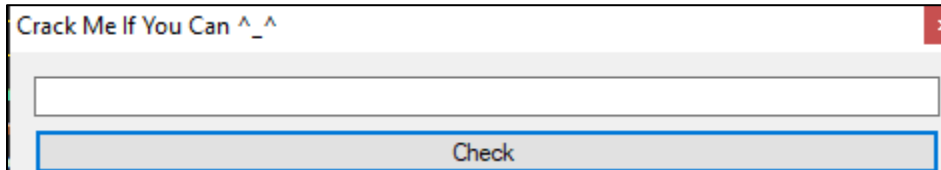


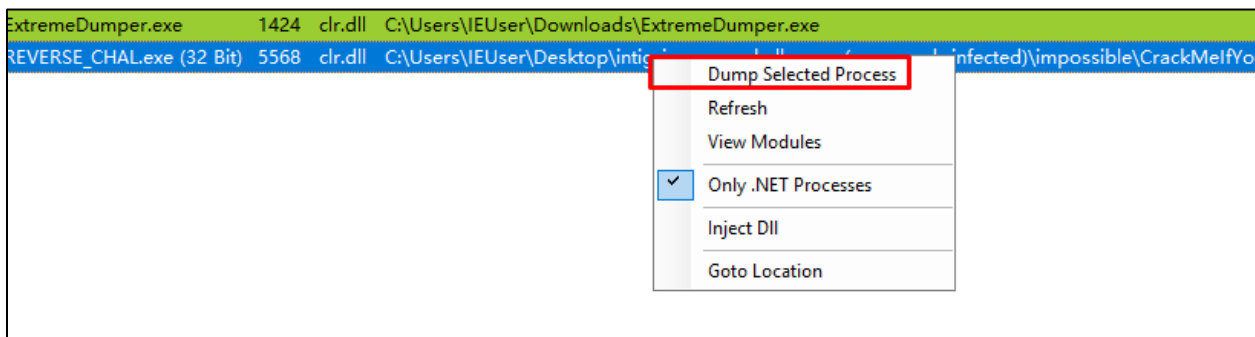
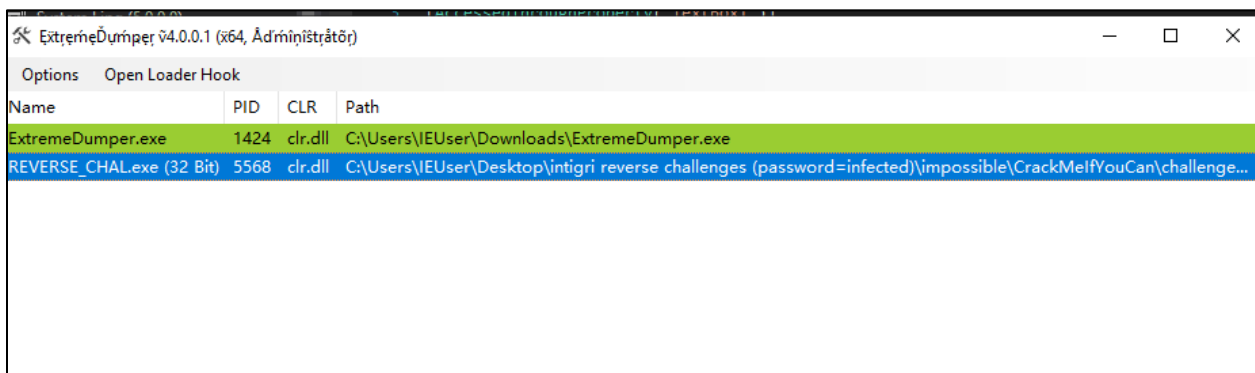
CrackMelfYouCan



The idea of this challenge is trying to deal with a packed and encrypted .Net

First thing we have to do is trying to unpack the .Net exe and dumping the unpacked exe so we can analyze the C# code. The program will have to unpack and decrypt itself to run correctly. So, we first run the program.



Then using a tool like Extreme dumper we extract the C# code.



 REVERSE_CHAL.dll	9/13/2023 4:42 AM	Application extens...	25 KB
 REVERSE_CHAL.exe	9/13/2023 4:42 AM	Application	23 KB

Now the code is unpacked and decrypted, we go to view what will happen when we click the check button.

```

1 // REVERSE_CHAL.Main
2 // Token: 0x06000025 RID: 37 RVA: 0x0000269C File Offset: 0x0000089C
3 private void Button1_Click(object sender, EventArgs e)
4 {
5     string @string = Encoding.UTF8.GetString(Resource.enc);
6     if (Operators.CompareString(this.unknownMethod(this.TextBox1.Text, @string), "WhatAreYouDoingToChallenge", false) == 0)
7     {
8         Interaction.MsgBox("You Solve It", MsgBoxStyle.Information, "Nice");
9     }
10 }

```

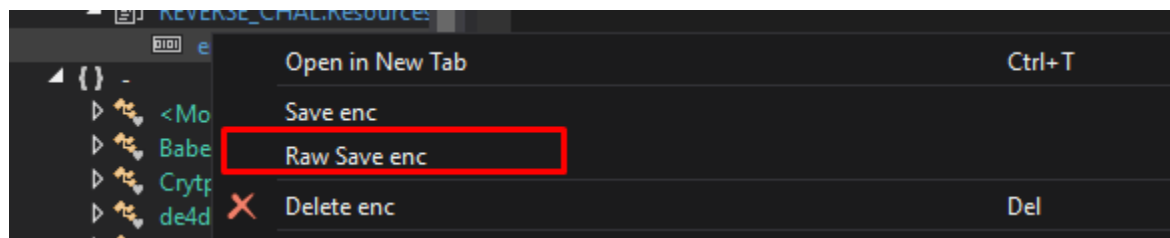
We find the input will through the method “unknownMethod” and the output will be compared to WhatAreYouDoingToChallenge.

```

1 // REVERSE_CHAL.Main
2 // Token: 0x06000026 RID: 38 RVA: 0x000026EC File Offset: 0x000008EC
3 public string unknownMethod(string textToScramble, string password)
4 {
5     StringBuilder stringBuilder = new StringBuilder(textToScramble.Length);
6     int num = 0;
7     checked
8     {
9         int num2 = textToScramble.Length - 1;
10        for (int i = num; i <= num2; i++)
11        {
12            int index = i % password.Length;
13            char c = textToScramble[i];
14            c = Strings.Chw((int)(c ^ password[index]));
15            stringBuilder.Append(c);
16        }
17        return stringBuilder.ToString();
18    }
19 }
20

```

We check the unknown method and see that it will take the input and xor it with the resource called enc.



So, we extract the raw hex of the resource file called enc

