## Info:

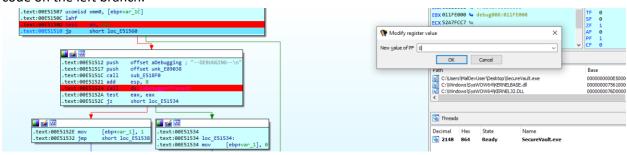
The Redteam found a seemingly very valuable binary. It looks like to be some sort of Password-Safe but we are missing the password

and need help to find a way into the vault. Can you help us?

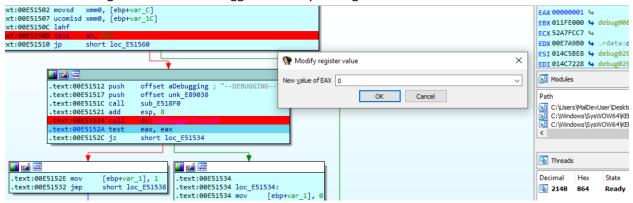
## Task:

## Find out what is inside thevault

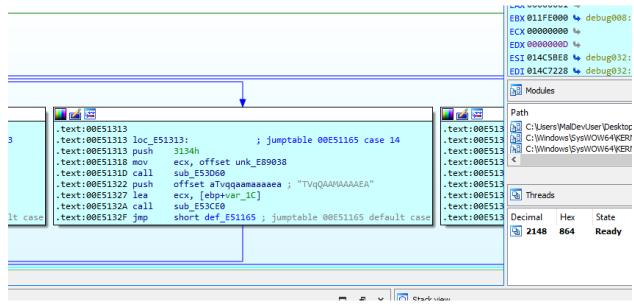
 Simply checking the binary it looks like we want to end up on the left side where "IsDebuggerPresent" is located. By changing the register PF we can force the program to run the code on the left branch.



2. Next we want to get around the Debugger-Check by setting eax to 0



3. As we do not know which path is correct and will yield the flag we try just stepping trough without changing anything. We do that until we end up at a bigger switch-case. The text in the console mentioned the key 14 so we want to try to select case 14. We can control the outcome of the switch-case by setting exd to 0xd



4. Stepping trough the rest of the code we can see the string "DEBUGGING INTERFACE" this seems like something we want to check out.

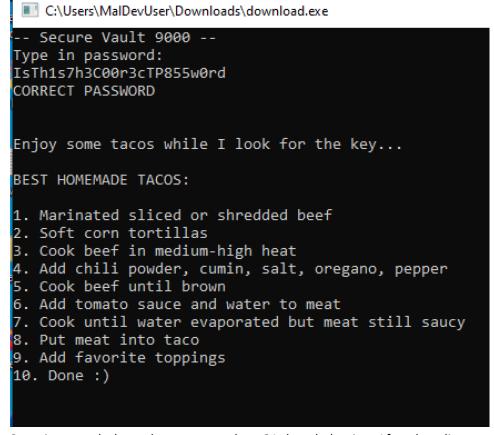
```
ecx, [eop+var_4c]
 rexr:nncotons tea
.text:00E513D5 push
                           ecx
.text:00E513D6 lea ecx, [ebp+var_34]
.text:00E513D9 call sub_E54220
.text:00E513DE push offset aDebuggingInter; "\nDEBUGGING INTERFACE:\nSECRET VAULT CO"...
.text:00E513E3 push offset unk_E89038
.text:00E513E8 call sub_E518F0
.text:00E513ED add esp, 8
.text:00E513F0 push offset asc E79B58 ; "\n"
.text:00E513F5 lea edx, [ebp+var 34]
.text:00E513F8 push edx
.text:00E513F9 lea
                          eax, [ebp+var_94]
.text:00E513FF push
                           eax
.text:00E51400 call
                           sub E51C70
```

5. After a few calls this will print out a very long code as "SECRET VAULT CODE"

- 6. Looking at the code this could be base64-encoded
- 7. Seeing the MZ-header we can be sure that this is another binary so we copy the whole code and decode it and then save it as dump.exe

hAGwARgBpAGwAZQBuAGEAbQBlAAAAcgBlAC0AMQAtAGUAYQBzAHkALgBlAHgAZQAAADQACgABAFAAcgBvAGQAdQBjAHQATgBhAG0AZQAAAAAACgBlAC0 AMQATAGUAYQBZAHKAAAA0AAgAAQBQAHIAbwBKAHUAYwB0AFYAZQByAHMAaQBvAG4AAAAXAC4AMAAUADAALgAwAAAOAAIAAEAQQBZAHMAZQBTAGIAbAB 5ACAAVgB1AHIAcwBpAG8AbgAAADEALgAwAC4AMAAuADAAAAC8QwAA6gEAAAAAAAAAAAAA77u /PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9InllcyI /Pg0KDQo8YXNzZW1ibHkgeG1sbnM9InVybjpzY2hlbWFzLW1pY3Jvc29mdC1jb206YXNtLnYxIiBtYW5pZmVzdFZlcnNpb249IjEuMCI+DQogIDxhc3N lbWJseUlkZW50aXR5IHZlcnNpb249IjEuMC4wLjAiIG5hbWU9Ik15QXBwbGljYXRpb24uYXBwIi8+DQogIDx0cnVzdEluZm8geG1sbnM9InVybjpzY2h lbWFzLW1pY3Jvc29mdC1jb206YXNtLnYyIj4NCiAgICA8c2VjdXJpdHk+DQogICAgICA8cmVxdWVzdGVkUHJpdmlsZWdlcyB4bWxucz0idXJuOnNjaGV tYXMtbWljcm9zb2Z0LWNvbTphc20udjMiPg0KICAgICAgICA8cmVxdwVzdGVkRXhlY3V0aW9uTGV2ZWwgbGV2ZWw9ImFzSW52b2tlciIgdwlBY2Nlc3M 9ImZhbHNlIi8+DQogICAgICA8L3JlcXVlc3RlZFByaXZpbGVnZXM+DQogICAgPC9zZWN1cml0eT4NCiAgPC90cnVzdEluZm8+DQo8L2Fzc2VtYmx5PgA mmc 14336 📻 1 **a** 🗇 🗗 🖸 Output 🎉 MZ•\\\\\ÿÿ\\ \\\\\@\\\\\\\\\ rogram cannot be run in

- 8. Checking the binary with DiE we can see that is a .NET-Binary
- 9. To debug .NET we can use dnSpy (32Bit in this case)
- 10. Simply stepping trough the binary we can figure out that the password for the binary is IsTh1s7h3C00r3cTP855w0rd



11. Stepping trough the code we get to a base64-decoded string. After decoding we can see the actual flag

```
uzwruihBoYXJIdHJNIGInZZ5NIGF];
string text3 = "delicious";
text3 = text2.Substring(1055, 36);
text3 = Encoding.UTF8.GetString(Convert.FromBase64String(text3));
string topping = "none";
46
100 % -
                                                     topping = args[0]:
Name
                                                                                             Value
                                                                                                                                                                                          Туре

    ▶ ✓ System.Text.Encoding.UTF8.get returned
    ▶ ⋄ System.Convert.FromBase64String returned
    ⋄ System.Text.Encoding.GetString returned

                                                                                             (System.Text.UTF8Encoding)
                                                                                                                                                                                        System.Te
                                                                                             (byte[0x0000001A])
                                                                                                                                                                                        byte[]
                                                                                             "DS{T4a7_wA$_pR337y_S1mP7e}"
string[0x00000001]
                                                                                                                                                                                        string[]
   text
                                                                                            "CORRECT PASSWORD"
   flag
   flag2
   e text2
                                                                                             "ZWxIaWZIbmQgZG9uZWMgcHJIdGI1bSB2dWxwdXRhdGUgc2FwaWVul..." {\it string}
                                                                                            "DS{T4a7_wA$_pR337y_S1mP7e}"
null
   text3
   topping
```