Write-up_picoCTF_2021

Chall: Checkpass

You can find out the source of challenge and script here(source, script)

In this challenge, there are an big table containing 1024 integer- elements. To easily solve it, I wrote IDA- Python script to get data from executable file, calculate and recover the original password.

Proceed to solve

1 - Let's pay for your attention to **sub_5960().** It's so exciting function and maybe contain main processing of program. For example, there are some functions such as **sub_66A0()**, **sub_6660()**, **sub_6600** which used to print notice to user about "**Success**", "**Invalid length\n**", "**Invalid password\n**" after entering the password.

```
    {
      *(_QWORD *)&v8 = 18LL;
      if ( v29 == byte_39D95[18] )
        sub_66A0();// func prints "Success"
    }
  }
}
```

```
272               }
273               sub_6650();                   // function print "Invalid password\n"
274             }
275             sub_6600();                     // Invalid length password
276           }
277           sub_34220(v2, 33LL, 0LL, 32LL, &off_248260);
278         }
279         sub_34220(v0, 41LL, 8LL, 41LL, &off_248260);
280       }
```

2 - As mentioned above, there are a big 1024 -element table at **.rodata:0000000000039560(** to be simple, also called T**),** it contains four 256-elements arrays internally. Additionally, there is also a permutation array(to be simple, also called P) at location **0x39970** including 128 elements(the size of its element is 8 bytes). It also contains 4 smaller -arrays.

3 - Your input as **0 generated array(inputArr0)**, after calling **sub_54E0()**, it returns **1st generated array**. The result array (1st generated array) is continuously as the argument for 2nd calling of **sub_54E0()**. This stage, in **sub_54E0()**, T and P will jump 2nd smaller array of them to evaluate with **1st generated array**. The end of **sub_54E0()**, function returns **2nd generated array.** Continuing as same.

```
v8 = v7;
sub_54E0((__int64)&v71, (unsigned __int8 *)&v8, 0LL);
v9 = v72;
v8 = v71;
sub_54E0((__int64)&v73, (unsigned __int8 *)&v8, 1LL);
v9 = v74;
v8 = v73;
sub_54E0((__int64)&v67, (unsigned __int8 *)&v8, 2LL);
v9 = v68;
v8 = v67;
sub_54E0((__int64)&v33, (unsigned __int8 *)&v8, 3LL);
v24 = v36;
v30 = v37;
v21 = v38;
v32 = v39;
```

4 - After totally 4 callings to sub_54E0(), the final generated array(4th generated array) will compared to target array located at **.rodata:0000000000039D95(** if check true, noticing to user: "Success")

Analyzing the sub_54E0() function

- To simplify for analyzing process, I will rename this function to ***HandleArray(outputArray, inputArray, x)***, where x indicate the index of smaller array in table T and permutation P. (x = [0, 3])
- Firstly, function will build **stackArr** on stack following bellow rule:
  - stackArr[i] = TableT[x + inputArr[i]]

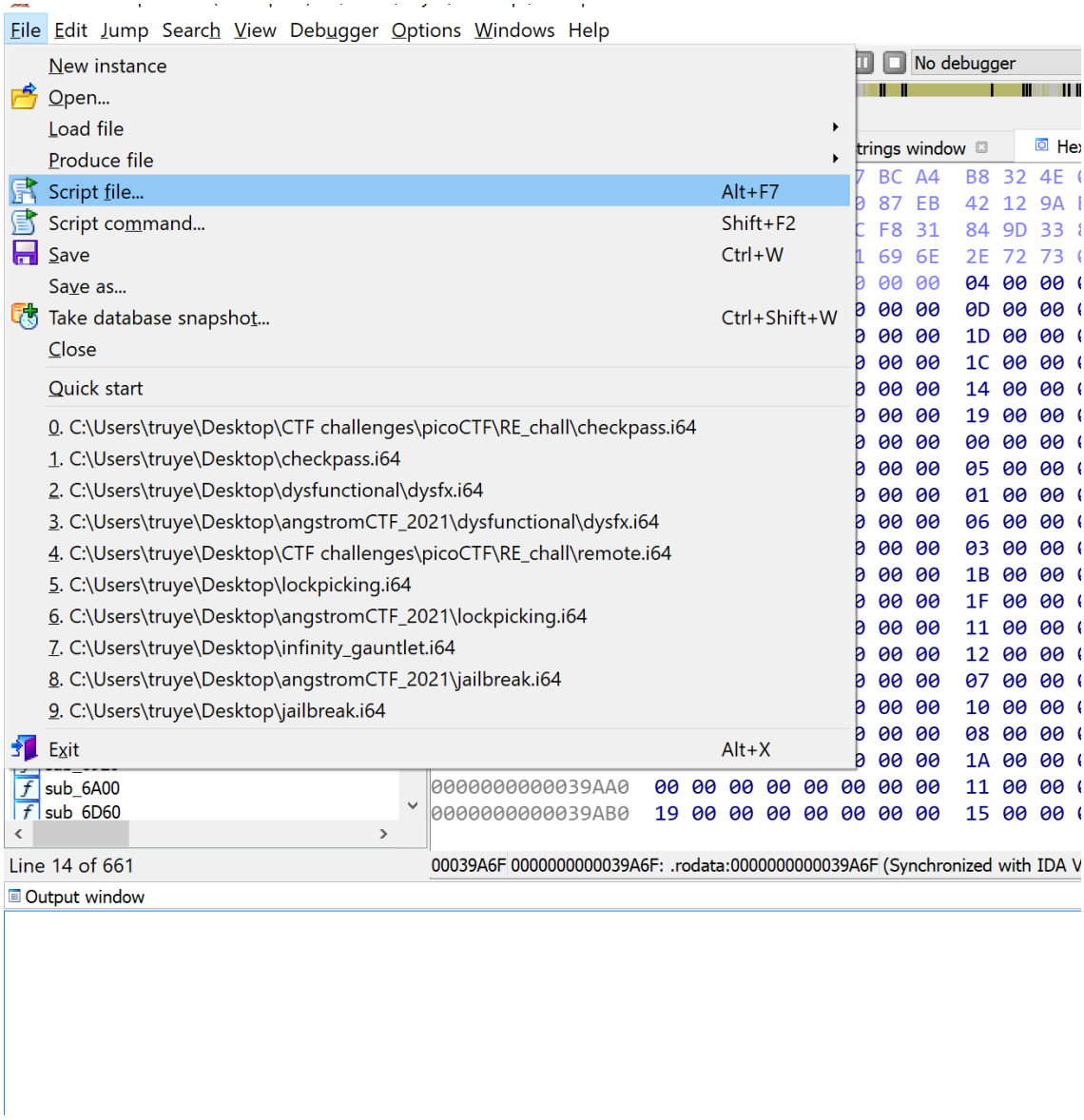
```
v6 = TableT[x + *inputArr];
v7 = TableT[x + inputArr[1]];
v8 = TableT[x + inputArr[2]];
v9 = TableT[x + inputArr[3]];
v10 = TableT[x + inputArr[4]];
v11 = TableT[x + inputArr[5]];
v12 = TableT[x + inputArr[6]];
v13 = TableT[x + inputArr[7]];
v14 = TableT[x + inputArr[8]];
v15 = TableT[x + inputArr[9]];
v16 = TableT[x + inputArr[10]];
v17 = TableT[x + inputArr[11]];
v18 = TableT[x + inputArr[12]];
v19 = TableT[x + inputArr[13]];
v20 = TableT[x + inputArr[14]];
```

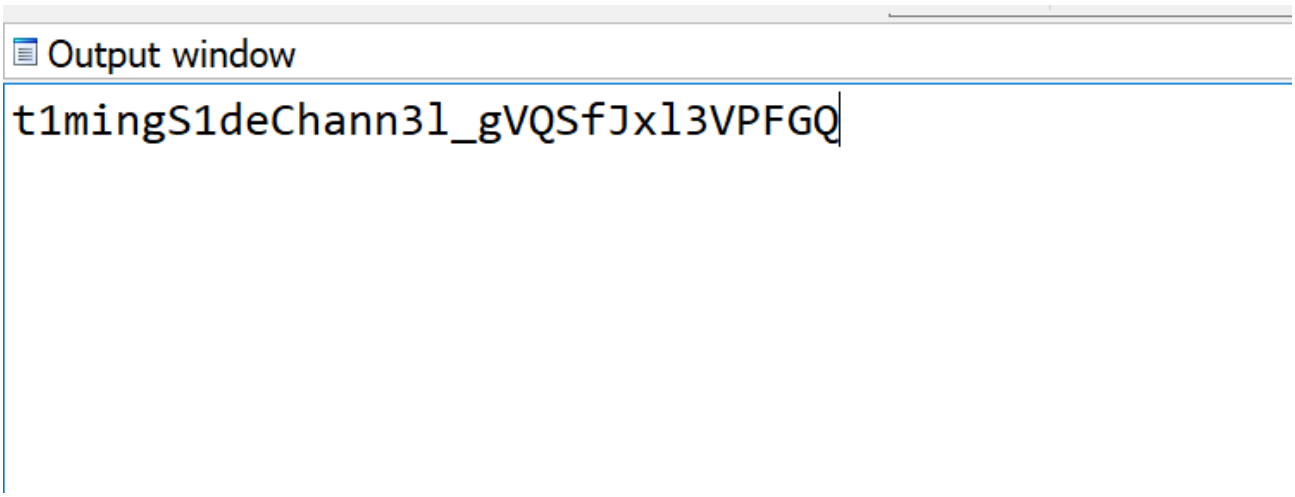
- Then, **outputArr**[**j**] = **stackArr**[**P**[**x + j**]].

Conclusion: From target array which is available from this challenge, I write script to recover the original array. After recovering, I will get the right password!

IDA-python Script(supported by IDA pro)

I will write the IDA-python script to get data from executable file and , then running that script with ***IDA-pro*** to print flag to IDA's output window:


File Edit Jump Search View Debugger Options Windows Help
New instance
Open...
Load file
Produce file
Script file... Alt+F7
Script command... Shift+F2
Save Ctrl+W
Save as...
Take database snapshot... Ctrl+Shift+W
Close
Quick start
0. C:\Users\truye\Desktop\CTF challenges\picoCTF\RE_chall\checkpass.i64
1. C:\Users\truye\Desktop\checkpass.i64
2. C:\Users\truye\Desktop\dysfunctional\dysfx.i64
3. C:\Users\truye\Desktop\angstromCTF_2021\dysfunctional\dysfx.i64
4. C:\Users\truye\Desktop\CTF challenges\picoCTF\RE_chall\remote.i64
5. C:\Users\truye\Desktop\lockpicking.i64
6. C:\Users\truye\Desktop\angstromCTF_2021\lockpicking.i64
7. C:\Users\truye\Desktop\infinity_gauntlet.i64
8. C:\Users\truye\Desktop\angstromCTF_2021\jailbreak.i64
9. C:\Users\truye\Desktop\jailbreak.i64
Exit Alt+X
No debugger
sub_6A00
sub_6D60
0000000000039AA0 00 00 00 00 00 00 00 00 11 00 00
0000000000039AB0 19 00 00 00 00 00 00 00 15 00 00
Line 14 of 661
00039A6F 0000000000039A6F: .rodata:0000000000039A6F (Synchronized with IDA V
Output window


Output window
t1mingS1deChann3l_gVQSfJxl3VPFGQ