toaster VIP_access_me

https://crackmes.one/crackme/5d0251c133c5d41c6d56e0c3

Crackme by b1h0 https://crackmes.one/user/b1h0

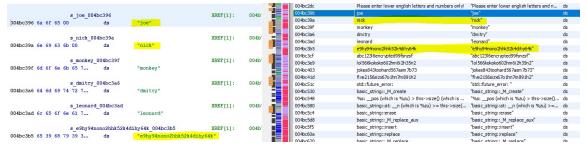
Date: 04/oct/2019

To analyze this file I have used two methods: static, with Ghidra, and dynamic with x64dbg. (In Ghidra there are few notes. I am still learning).

Ghidra: Some useful notes

 After analyzing the binary, if we look at the strings in memory it is easy to identify what the user names would be, and probably encrypted vip codes.

User names and encrypted passwords strings



x64dbg: Some useful notes

- · Main function is located at 00401000
- At the beginning of this subroutine, it sets a value of 1 over a memory position, where at the end of the process it returns to 0, so it is apparently a Boolean control variable.

Some type of control variable

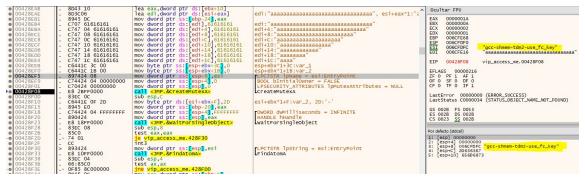


At address 006CFE16 fills char 'a' 64 times.

04/10/2019 B14ck H013

Create a Mutex using string "gcc-shmem-tdm2-use_fc_key".
 This information is probably not relevant to find the VIP CODES, but I liked to put it. :sweat smile:

Mutex creation



- At address 006CFDFC, after the creation of Mutex, to the string used are added all the
 'a' that had been created previously, but changing some lowercase 'a' to uppercase 'A'.
 Remaining the resulting string as follows: "gcc-shmem-tdm2-use_fc_keyaaaaaaaaAAaaAaaAaaAAAAAaaaa"
- At address 004B8080 It makes a comparison of different memory positions and activates a flag (Boolean variable) for each memory position that has the value 1 activated that is also supposedly Boolean. The same at address 004B7FF0 & 004B7F20 & 004B7E50 (other information that is not useful to solve)

A sub that do memory comparisons and activate boolean flag

```
004B807F
      004B8080
                                        803D 50B14B00 00
75 07
                                                                                                       cmp byte ptr ds:[4BB150],0
                                                                                                                                                                                           do_comparisons_and_flag
                                                                                                     cmp byte ptr ds:[4B8150],0
jne vip_access_me.4B8090
mov byte ptr ds:[4B8150],1
cmp byte ptr ds:[4B8158],0
jne vip_access_me.4B80A0
mov byte ptr ds:[4B8158],1
cmp byte ptr ds:[4B81A8],0
jne vip_access_me.4B80B0
mov byte ptr ds:[4B81A8],1
cmp byte ptr ds:[4B81A8],0
jne vip_access_me.4B80C0
                                        C605 50B14B00 01
803D 58B14B00 00
      004B8089
      004B8090
                                  75 07

C605 58814800 00

> 803D A8814800 00

75 07

C605 A8814800 01
       004B8097
      004R8099
       004B80A0
      004B80A7
004B80A9
                               > 803D B8B14B00 00

> 75 07
      004B80B0
004B80B7
                                                                                                     cmp byte ptr ds:[4BB1B8],0
jne vip_access_me.4B80C0
mov byte ptr ds:[4BB188],1
cmp byte ptr ds:[4BB188],0
jne vip_access_me.4B80D0
mov byte ptr ds:[4BB188],0
jne vip_access_me.4B80E0
mov byte ptr ds:[4BB198],0
jne vip_access_me.4B80E0
mov byte ptr ds:[4BB198],1
cmp byte ptr ds:[4BB198],0
jne vip_access_me.4B80E0
       004B80B9
      004B80C0
004B80C7
                               > 803D 88B14B00 00

• 75 07
                               C605 88B14B00 01

> 803D 98B14B00 00

75 07

C605 98B14B00 01
      004B80C9
004B80D0
       004B80D7
      004B80D9
                               > 803D 78B14B00 00

> 75 07

C605 78B14B00 01
       004B80E0
                                                                                                      jne vip_access_me.4B80F0
mov byte ptr ds:[4BB178],1
cmp byte ptr ds:[4BB168],0
      004B80E7
       004B80E9
      004B80F0
004B80F7
                                        803D 68B14B00 00
75 07
-0
                                                                                                                                             me.4B810
                                                                                                       mov byte ptr ds:[4BB168],1
       004B80F9
                                        C605 68B14B00 01
       004B8100
                                        F3:C3
                                                                                                       ret
```

• From the address **004B769F** are the functions that the user names reveal to us. The names are: **joe**, **nick**, **monkey**, **dmitry** and **leonard**.

B14ck H0l3 04/10/2019

Username strings

```
Strings

83EC 04

C70424 90144000

88 159CF4FF
B9 F8904C00

C70424 96534B00

88 D49EF4FF
83EC 04

C70424 70144000

88 F59BF4FF
B9 E0904C00

C70424 9AC34B00

88 B49EF4FF
83EC 04

C704224 9C144000

88 D59BF4FF
B9 C8904C00

C70424 9F634B00

88 949EF4FF
89 C8904C00

C70424 30144000

88 B59BF4FF
B9 B9904C00

C70424 AC34B00

88 749EF4FF
89 B0904C00

C70424 AC34B00

88 559BF4FF
B9 B9904C00

C70424 AC34B00

88 559BF4FF
B9 B9904C00

C70424 ADC34B00

88 559BF4FF
B9 B9904C00

C70424 ADC34B00

88 549EF4FF
B9 S8904C00

C70424 ADC34B00

88 549EF4FF
B9 S8904C00

C70424 ADC34B00

88 549EF4FF
                                                                                 sub esp,4
mov dword ptr ss:[esp],<vip_access_me.sub_40149
call <vip_access_me.sub_4012c0>
                                                                                                                                                                                                                                                                Arg1 = <vip_access_me.sub_401490>
Lsub_4012c0
                                                                               mov dwore pt.

call svip_access_me.sub_4012CUs

mov ecx,vip_access_me.4090F8

mov dword ptr ss: [esp]_vvip_access_me.users>
call svip_access_me.sub_401590s

sub esp, 4

mov dword ptr ss: [esp]_vvip_access_me.sub_401470>
call svip_access_me.sub_4012COs

mov ecx,vip_access_me.sub_4012COs

mov ecx,vip_access_me.4090E0

mov dword ptr ss: [esp]_vip_access_me.4BC39A

call svip_access_me.sub_401590s

cub_asp.4
                                                                                                                                                                                                                                                                 Arg1 = "joe
sub_401590
                                                                                                                                                                                                                                                                [Arg1 = <vip_access_me.sub_401470>
sub_4012c0
                                                                                                                                                                                                                                                                Arg1 = "nick
sub_401590
                                                                                  Arg1 = <vip_access_me.sub_401450>
Lsub_4012C0
                                                                                                                                                                                                                                                               2008:&"monkey"
| Arg1 = "monkey"
| sub_401590
                                                                                   mov ecx.vip_access_me.4C90C8
mov dword ptr ss:[esp],vip_access_me.4BC39F
call <vip_access_me.sub_401590>
                                                                                 call svip_access_me.sub_401590s
sub_esp,4
mov_dword_ptr_ss:[esp],<vip_access_me.sub_401430>
call svip_access_me.sub_4012C0s
mov_ecx_vip_access_me.sub_4012C0s
                                                                                                                                                                                                                                                                Arg1 = "dmitry"
Lsub_401590
                                                                                   mov ecx,vip_access_me.4e0080
mov dword ptr ss:[esp],vip_access_me.4BC3A6
call <vip_access_me.sub_401590>
                                                                               call vip_access_me.sub_40190s
sub_esp,4
mov_dword_tr_ss:[esp]_vip_access_me.sub_401410>
call_evip_access_me.sub_4012C0s
mov_ecx_vip_access_me.49098
mov_dword_tr_ss:[esp]_vip_access_me.48C3AD
call_evip_access_me.sub_401590s
                                                                                                                                                                                                                                                                [Arg1 = <vip_access_me.sub_401410>
                                                                                                                                                                                                                                                              Lsub_401590
```

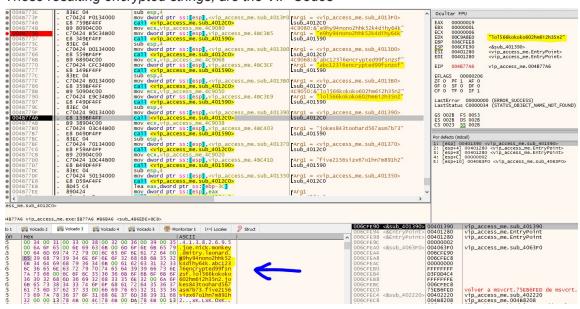
At 004B4160 there is the function that compares the string with the username.

Compare Username



 From address 004B7750 some operations with strings like: e9hy94nono2hhk52k4dihy64k, abc123l6encrypted99fsnzsf. They are the "VIP codes" that are encrypted.

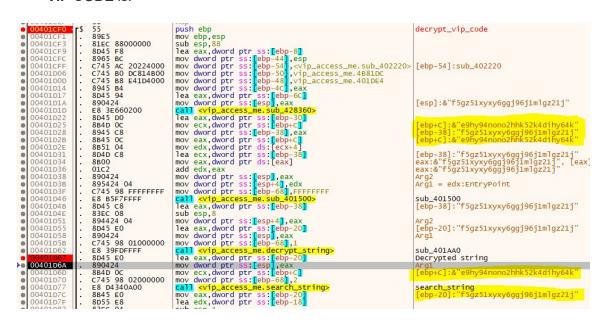
These resulting encrypted strings, are the VIP Codes



· At address 004B6400 Ask for the data

B14ck H0l3 04/10/2019

And now, the function that really matters to us. The one that decrypts the VIP Code. It
is in the address 00401CF0. Just put a breakpoint at address 00401D67 and check that
it is what is loaded into the EAX register that contains the memory address where the
VIP CODE is.



Here are the list of names, ecoded and decoded vip codes are:

94nono2hhk52k4dihy64k	f5gz51xyxy6ggj96j1mlgz21j
.2316encrypted99fsnzsf	bac468i2fxctzvrfm55eqx0qe
666kokoko602hm6i2h35n2	iyi922jyjyjy206gd216g89x6
es843toohard567asm7b73	kyjfq318ryygbtm927bqd7a78
2156sizx67o1hn7m891h2	elwf6492q10n27y4gx7d354g6
-	2316encrypted99fsnzsf 66kokoko602hm6i2h35n2 s843toohard567asm7b73

B14ck H0l3 04/10/2019

But something curious happens. Only work VIP Codes for *joe* and *monkey* users. The other VIP Codes that are generated when tested do not seem correct as the following message is displayed: "Please enter lower english letters and numbers only!"

Result of the two VIP Codes that work.

VIP Codes that do not work: "Please enter lower english letters and numbers only!"

```
CrackMe\crackmes.one\toaster-VIP_access_me>VIP_access_me.exe
Are you a vip member? (y/n): y
What is your name?: nick
Please enter your vip code: bac468i2fxctzvrfm55eqx0qe
Please enter lower english letters and numbers only!
Presione una tecla para continuar . . .
```

That's all folks!!!

B14ck H0l3 04/10/2019