

Lab 2 GDB

Author: Anastasiia Stepanova

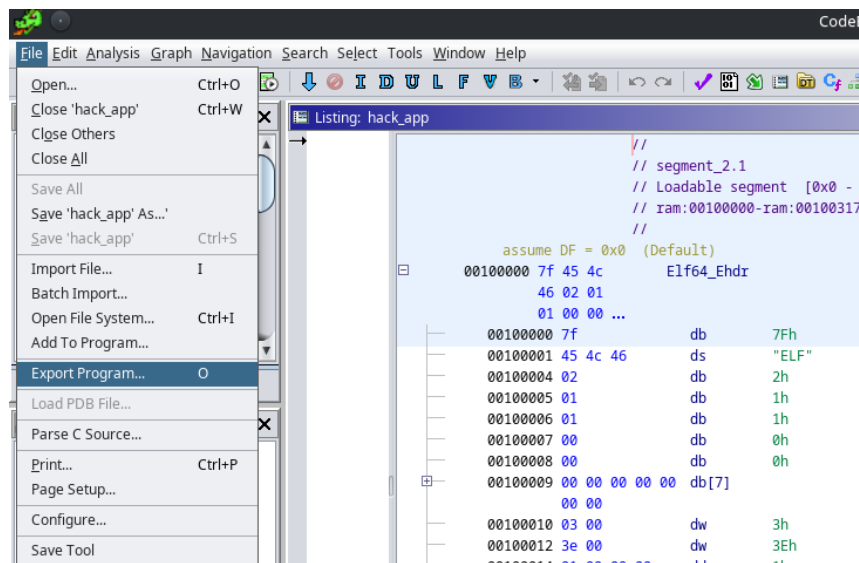
1. Create keygen for hack_app.
2. Create binary patch to disable licensing completely.

Solution:

-

Ghidra

While analyzing the C code generated by the program, I found that the entire security logic is based on if statements (in the main function), which are easy to hack inside the binary.



```

42 if (iVar1 == 0) {
43     local_24 = 1;
44 }
45 if (local_24 == 0) {
46     printf("Your HWID is %08X%08X.\nEnter the license key:
47     __isoc99_scanf(&DAT_0010208f,userInput);
48     iVar1 = strcmp(md5decode,userInput,0x21);
49     if (iVar1 == 0) {
50         setxattr(binaryPath,"user.license",md5decode,0x21,0)
51         puts("Now you app is activated! Thanks for purchasin
52     }
53     else {
54         puts("Provided key is wrong! App is closing!");
55     }
56 }
57 else if (local_24 == 1) {
58     puts("Your app is licensed to this PC!");
59 }
60 system("read -p \"Press Enter to continue...\" var");
61 if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
62     /* WARNING: Subroutine does not return

```

Example:

1. To change result of the code if (iVar1==0) (the variable is used to check 1- is the license already provided AND is the license key right) represented in assembler as 75 07 Where 75 is code of **JNZ** command, all is needed is to change JNZ to JZ by changing 75 to 74.

The result will be the following: 74 07 or in C code if (iVar != 0). Operations description from StackOverflow “{ Op Code | mnemonic | Description ————|—————|————— 74 cb | JE rel8 | Jump short if equal (ZF=1). 74 cb | JZ rel8 | Jump short if zero (ZF ← 1).

0F 84 cw	JE rel16	Jump near if equal (ZF=1). Not supported in 64-bit mode.
0F 84 cw	JZ rel16	Jump near if 0 (ZF=1). Not supported in 64-bit mode.
0F 84 cd	JE rel32	Jump near if equal (ZF=1).
0F 84 cd	JZ rel32	Jump near if 0 (ZF=1).

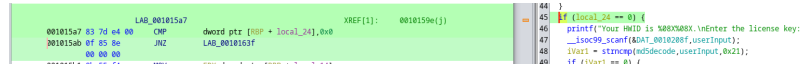
```

75 cb      | JNE rel8 | Jump short if not equal (ZF=0).
75 cb      | JNZ rel8 | Jump short if not zero (ZF=0).

0F 85 cd   | JNE rel32 | Jump near if not equal (ZF=0).
0F 85 cd   | JNZ rel32 | Jump near if not zero (ZF=0).}

```

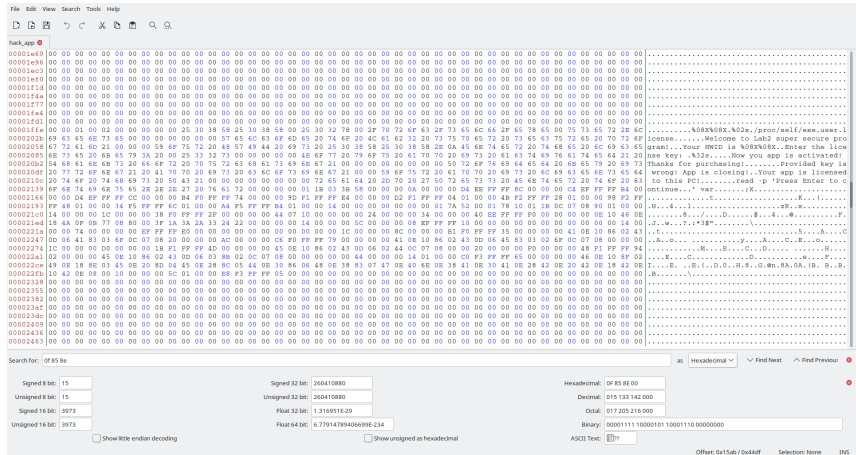
2. The second approach is to change the value in CMP function, when a variable is comparing with constant. I did that for local_24==0.



I changed the binary line 83 7d e4 00 by 83 7d e4 01. The result code looks like `if (local_24 == 1)`

Bless

I didn't figure out how to modify the binary inside Ghidra and just wanted to keep the original C-described program, so I used Bless to modify the binary. So, I used both of the hacks above to generate the key in the same way as I did to create the patch.

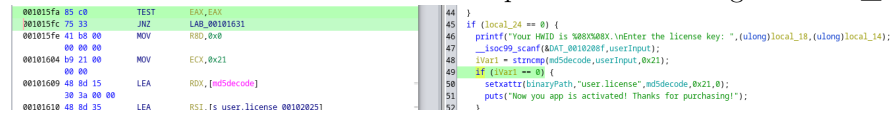


Keygen

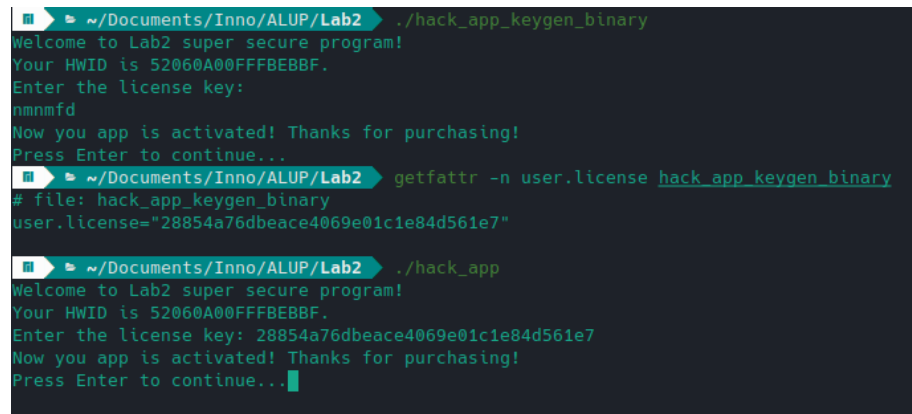
After looking at the binary for several hours, I realized that I did not want to go so deep into the assembly and logic of the functions of external libraries called to create the md5decode variable.

Therefore, the easiest way for me was to make the program think that I passed the correct key as in the example 1. After that I sim-

ply intercept the license of the patched program via `getfattr -n user.license`. The license is then placed in the original `hack_app`.



Result: 1 - we get the key 2 - the key is working



```

~/Documents/Inno/ALUP/Lab2 ➤ ./hack_app_keygen_binary
Welcome to Lab2 super secure program!
Your HWID is 52060A00FFBEBBF.
Enter the license key:
nmnmfd
Now you app is activated! Thanks for purchasing!
Press Enter to continue...

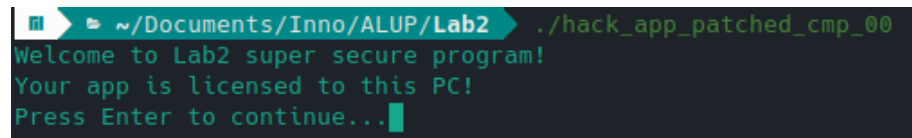
~/Documents/Inno/ALUP/Lab2 ➤ getfattr -n user.license hack_app_keygen_binary
# file: hack_app_keygen_binary
user.license="28854a76dbeace4069e01c1e84d561e7"

~/Documents/Inno/ALUP/Lab2 ➤ ./hack_app
Welcome to Lab2 super secure program!
Your HWID is 52060A00FFBEBBF.
Enter the license key: 28854a76dbeace4069e01c1e84d561e7
Now you app is activated! Thanks for purchasing!
Press Enter to continue...

```

Patch

I used the logic described in the second example to go directly to the line “your application is licensed for this PC!”.



```

~/Documents/Inno/ALUP/Lab2 ➤ ./hack_app_patched_cmp_00
Welcome to Lab2 super secure program!
Your app is licensed to this PC!
Press Enter to continue...

```