Lab 2 GDB

Author: Anastasiia Stepanova Sourse ASSEMBLY patch and keygen code: Available on Github

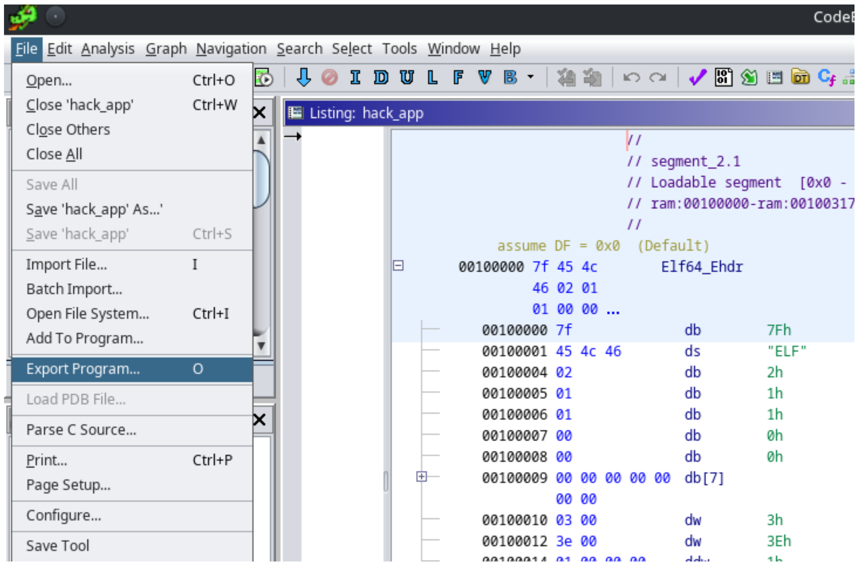
1. Create keygen for hack_app.
2. Create binary patch to disable licensing completely.

Solution:

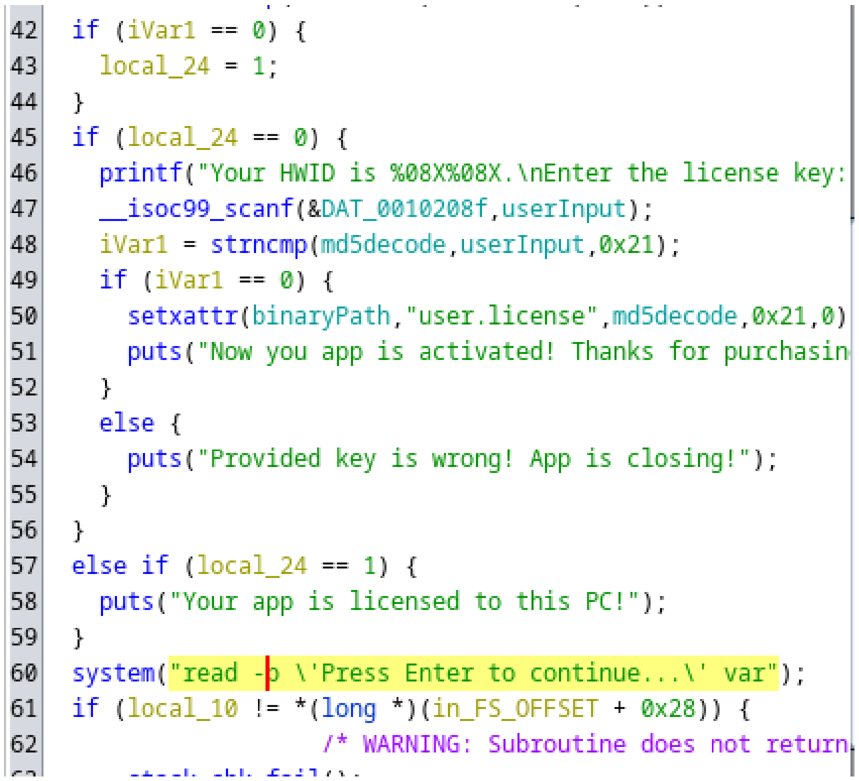
-

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.



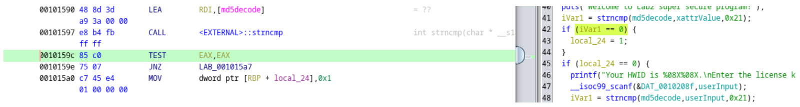
```
                        //
                        // segment_2.1
                        // Loadable segment  [0x0 -
                        // ram:00100000-ram:00100317
                        //
          assume DF = 0x0  (Default)
   00100000 7f 45 4c        Elf64_Ehdr
            46 02 01
            01 00 00 ...
     00100000 7f             db        7Fh
     00100001 45 4c 46       ds        "ELF"
     00100004 02             db        2h
     00100005 01             db        1h
     00100006 01             db        1h
     00100007 00             db        0h
     00100008 00             db        0h
     00100009 00 00 00 00 00 db[7]
              00 00
     00100010 03 00          dw        3h
     00100012 3e 00          dw        3Eh
```



```
if (iVar1 == 0) {
  local_24 = 1;
}
if (local_24 == 0) {
  printf("Your HWID is %08X%08X.\nEnter the license key:
  __isoc99_scanf(&DAT_0010208f,userInput);
  iVar1 = strncmp(md5decode,userInput,0x21);
  if (iVar1 == 0) {
    setxattr(binaryPath,"user.license",md5decode,0x21,0)
    puts("Now you app is activated! Thanks for purchasin
  }
  else {
    puts("Provided key is wrong! App is closing!");
  }
}
else if (local_24 == 1) {
  puts("Your app is licensed to this PC!");
}
system("read -p \'Press Enter to continue...\' var");
if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                  /* 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.

   ```
   00101590 48 8d 3d      LEA      RDI,[md5decode]                        = ??
            a9 3a 00 00
   00101597 e8 b4 fb      CALL     <EXTERNAL>::strncmp                    int strncmp(char * __s1
            ff ff
   0010159c 85 c0         TEST     EAX,EAX
   0010159e 75 07         JNZ      LAB_001015a7
   001015a0 c7 45 e4      MOV      dword ptr [RBP + local_24],0x1
            01 00 00 00
   ```

   ```
   puts("Welcome to Lab2 super secure program!");
   iVar1 = strncmp(md5decode,xattrValue,0x21);
   if (iVar1 == 0) {
     local_24 = 1;
   }
   if (local_24 == 0) {
     printf("Your HWID is %08X%08X.\nEnter the license k
     __isoc99_scanf(&DAT_0010208f,userInput);
     iVar1 = strncmp(md5decode,userInput,0x21);
   ```

   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).

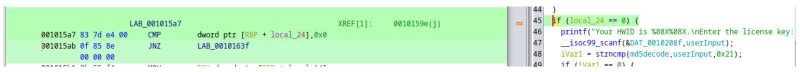
   ```
   OF 84 cw   | JE rel16  | Jump near if equal (ZF=1). Not supported in 64-bit mode.
   OF 84 cw   | JZ rel16  | Jump near if 0 (ZF=1). Not supported in 64-bit mode.

   OF 84 cd   | JE rel32  | Jump near if equal (ZF=1).
   OF 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`.

   ```
                    LAB_001015a7                                    XREF[1]:     0010159e(j)
   001015a7 83 7d e4 00     CMP        dword ptr [RBP + local_24],0x0
   001015ab 0f 85 8e        JNZ        LAB_0010163f
            00 00 00
   ```

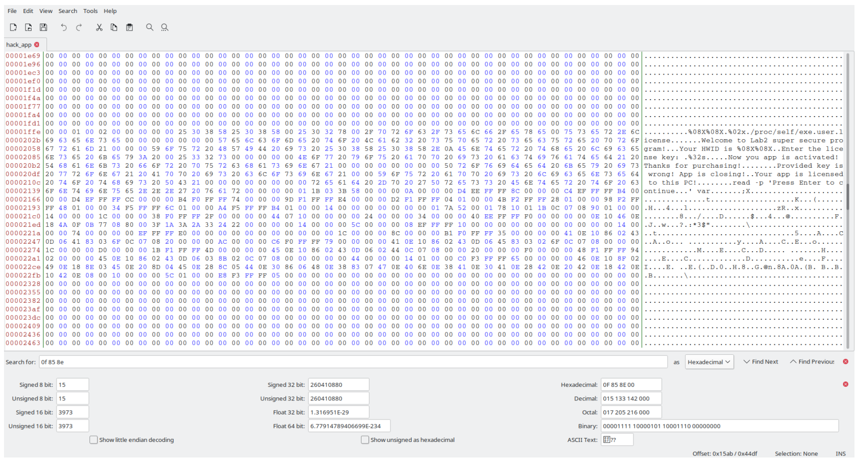
   ```
   }
   if (local_24 == 0) {
     printf("Your HWID is %08X%08X.\nEnter the license key:
     __isoc99_scanf(&DAT_0010208f,userInput);
     iVar1 = strncmp(md5decode,userInput,0x21);
     if (iVar1 == 0) {
   ```

   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.



```
00001e69 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001e96 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001ec3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001ef0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001f1d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001f4a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001f77 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001fa4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001fd1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00001ffe 00 00 01 00 02 00 00 00 00 00 25 30 38 58 25 30 38 58 00 25 30 32 78 00 2F 70 72 6F 63 2F 73 65 6C 66 2F 65 78 65 00 75 73 65 72 2E 6C .........%08X%08X.%02x./proc/self/exe.user.l
0000202b 69 63 65 6E 73 65 00 00 00 00 00 00 00 57 65 6C 63 6F 6D 65 20 74 6F 20 4C 61 62 32 20 73 75 70 65 72 20 73 65 63 75 72 65 20 70 72 6F icense.......Welcome to Lab2 super secure pro
00002058 67 72 61 6D 21 00 00 00 59 6F 75 72 20 48 57 49 44 20 69 73 20 25 30 38 58 25 30 38 58 2E 0A 45 6E 74 65 72 20 74 68 65 20 6C 69 63 65 gram!...Your HWID is %08X%08X..Enter the lice
00002085 6E 73 65 20 6B 65 79 3A 20 00 25 33 32 73 00 00 00 00 4E 6F 77 20 79 6F 75 20 61 70 70 20 69 73 20 61 63 74 69 76 61 74 65 64 21 20 nse key: .%32s.....Now you app is activated! 
000020b2 54 68 61 6E 6B 73 20 66 6F 72 20 70 75 72 63 68 61 73 69 6E 67 21 00 00 00 00 00 00 00 00 50 72 6F 76 69 64 65 64 20 6B 65 79 20 69 73 Thanks for purchasing!........Provided key is
000020df 20 77 72 6F 6E 67 21 20 41 70 70 20 69 73 20 63 6C 6F 73 69 6E 67 21 00 00 59 6F 75 72 20 61 70 70 20 69 73 20 6C 69 63 65 6E 73 65 64  wrong! App is closing!..Your app is licensed
0000210c 20 74 6F 20 74 68 69 73 20 50 43 21 00 00 00 00 00 00 00 00 72 65 61 64 20 2D 70 20 27 50 72 65 73 73 20 45 6E 74 65 72 20 74 6F 20 63  to this PC!........read -p 'Press Enter to c
00002139 6F 6E 74 69 6E 75 65 2E 2E 2E 27 20 76 61 72 00 00 00 00 01 1B 03 3B 58 00 00 00 0A 00 00 00 D4 EE FF FF 8C 00 00 00 C4 EF FF FF B4 00 ontinue...' var......;X.....................
00002166 00 00 D4 EF FF FF CC 00 00 00 B4 F0 FF FF 74 00 00 00 9D F1 FF FF E4 00 00 00 D2 F1 FF FF 04 01 00 00 4B F2 FF FF 28 01 00 00 98 F2 FF ..............t.................K...(......
00002193 FF 48 01 00 00 34 F5 FF FF 6C 01 00 00 A4 F5 FF FF B4 01 00 00 14 00 00 00 00 00 00 00 01 7A 52 00 01 78 10 01 1B 0C 07 08 90 01 00 00 .H...4...l..................zR..x.........
000021c0 14 00 00 00 1C 00 00 00 38 F0 FF FF 2F 00 00 00 00 44 07 10 00 00 00 00 24 00 00 00 34 00 00 00 40 EE FF FF F0 00 00 00 00 0E 10 46 0E ........8.../....D......$...4...@.........F.
000021ed 18 4A 0F 0B 77 08 80 00 3F 1A 3A 2A 33 24 22 00 00 00 00 14 00 00 00 5C 00 00 00 08 EF FF FF 10 00 00 00 00 00 00 00 00 00 00 00 14 00 .J..w...?.:*3$".......\.....................
0000221a 00 00 74 00 00 00 00 EF FF FF E0 00 00 00 00 00 00 00 00 00 00 00 1C 00 00 00 8C 00 00 00 B1 F0 FF FF 35 00 00 00 00 41 0E 10 86 02 43 ..t............................5....A....C
00002247 0D 06 41 83 03 6F 0C 07 08 20 00 00 00 AC 00 00 00 C6 F0 FF FF 79 00 00 00 00 41 0E 10 86 02 43 0D 06 45 83 03 02 6F 0C 07 08 00 00 00 ..A..o... ..........y....A....C..E...o......
00002274 1C 00 00 00 D0 00 00 00 1B F1 FF FF 4D 00 00 00 00 45 0E 10 86 02 43 0D 06 02 44 0C 07 08 00 00 20 00 00 00 F0 00 00 00 48 F1 FF FF 94 ...........M....E....C...D..... .......H....
000022a1 02 00 00 00 45 0E 10 86 02 43 0D 06 03 8B 02 0C 07 08 00 00 00 00 00 44 00 00 00 14 01 00 00 C0 F3 FF FF 65 00 00 00 00 46 0E 10 8F 02 ....E....C............D..........e....F....
000022ce 49 0E 18 8E 03 45 0E 20 8D 04 45 0E 28 8C 05 44 0E 30 86 06 48 0E 38 83 07 47 0E 40 6E 0E 38 41 0E 30 41 0E 28 42 0E 20 42 0E 18 42 0E I....E. ..E.(..D.0..H.8..G.@n.8A.0A.(B. B..B.
000022fb 10 42 0E 08 00 10 00 00 00 5C 01 00 00 E8 F3 FF FF 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .B.......\...................................
00002328 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00002355 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00002382 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
000023af 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
000023dc 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00002409 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00002436 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
00002463 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .............................................
```

Search for: 0f 85 8e as Hexadecimal

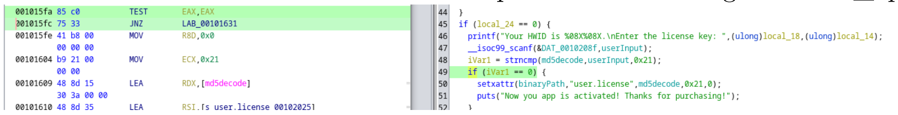
Signed 8 bit:	15	Signed 32 bit:	260410880	Hexadecimal:	0F 85 8E 00
Unsigned 8 bit:	15	Unsigned 32 bit:	260410880	Decimal:	015 133 142 000
Signed 16 bit:	3973	Float 32 bit:	1.316951E-29	Octal:	017 205 216 000
Unsigned 16 bit:	3973	Float 64 bit:	6.77914789406699E-234	Binary:	00001111 10000101 10001110 00000000
☐ Show little endian decoding		☐ Show unsigned as hexadecimal		ASCII Text:	??

Offset: 0x15ab / 0x44df   Selection: None   INS

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.

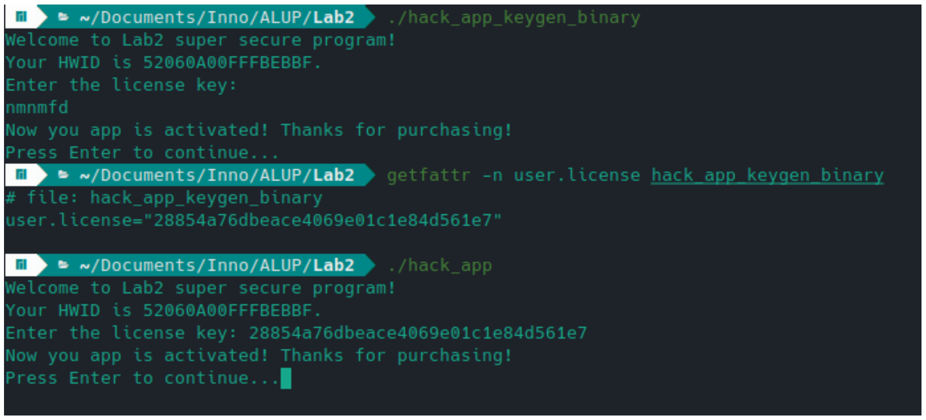
Therefore, the easiest way for me was to make the program think that I passed the correct key as in the example 1. After that I simply intercept the license of the patched program via **`getfattr -n user.license`**. The license is then placed in the original hack_app.



```
001015fa 85 c0        TEST    EAX,EAX
001015fc 75 33        JNZ     LAB_00101631
001015fe 41 b8 00     MOV     R8D,0x0
         00 00 00
00101604 b9 21 00     MOV     ECX,0x21
         00 00
00101609 48 8d 15     LEA     RDX,[md5decode]
         30 3a 00 00
00101610 48 8d 35     LEA     RSI,[s_user.license_00102025]
```

```
}
if (local_24 == 0) {
  printf("Your HWID is %08X%08X.\nEnter the license key: ",(ulong)local_18,(ulong)local_14);
  __isoc99_scanf(&DAT_0010208f,userInput);
  iVar1 = strncmp(md5decode,userInput,0x21);
  if (iVar1 == 0) {
    setxattr(binaryPath,"user.license",md5decode,0x21,0);
    puts("Now you app is activated! Thanks for purchasing!");
  }
```

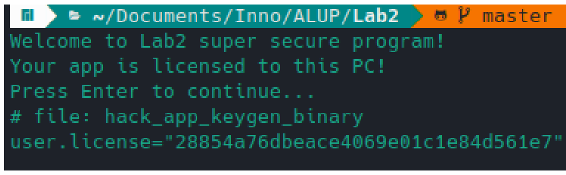
Result: 1 - we get the key 2 - the key is working



```
~/Documents/Inno/ALUP/Lab2  ./hack_app_keygen_binary
Welcome to Lab2 super secure program!
Your HWID is 52060A00FFFBEBBF.
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 52060A00FFFBEBBF.
Enter the license key: 28854a76dbeace4069e01c1e84d561e7
Now you app is activated! Thanks for purchasing!
Press Enter to continue...
```

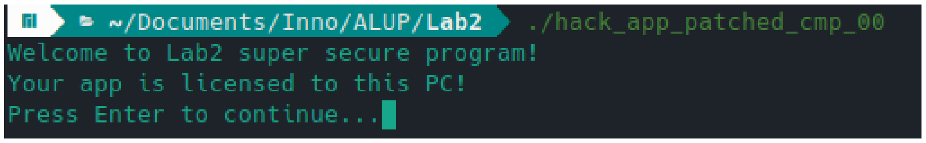
All the commands has been added to small bash **keygen** script.



```
~/Documents/Inno/ALUP/Lab2   master
Welcome to Lab2 super secure program!
Your app is licensed to this PC!
Press Enter to continue...
# file: hack_app_keygen_binary
user.license="28854a76dbeace4069e01c1e84d561e7"
```

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...
```