IU-Advanced-Linux-Assignment-2

Firstly, I just ran executable file:

```
admin@debian-vm:~/IU-advanced-linux/lab-2$ ./hack_app
Welcome to Lab2 super secure program!
Your HWID is A0060600FFFB8B17.
Enter the license key: 123
Provided key is wrong! App is closing!
Press Enter to continue...
```

So we see that there some validation based on our HWID, which we do not yet know where to get it, let's see what shows `strace ./hack_app`

```
mmap(0x7fcad275b000, 593920, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22e000) = 0x7fcad275b000
mmap(0x7fcad27ec000, 204800, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2be000) = 0x7fcad27ec000
mmap(0x7fcad281e000, 16336, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fcad281e000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@>\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1901536, ...}) = 0
mmap(NULL, 1914496, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fcad2359000
mmap(0x7fcad237b000, 1413120, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7fcad237b000
mmap(0x7fcad24d4000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17b000) = 0x7fcad24d4000
mmap(0x7fcad2523000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c9000) = 0x7fcad2523000
mmap(0x7fcad2529000, 13952, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fcad2529000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0000\21\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=18688, ...}) = 0
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fcad2353000
mmap(0x7fcad2354000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fcad2354000
mmap(0x7fcad2356000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fcad2356000
mmap(0x7fcad2357000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fcad2357000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 l\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=149520, ...}) = 0
mmap(NULL, 136304, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fcad2331000
mmap(0x7fcad2337000, 65536, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fcad2337000
mmap(0x7fcad2347000, 24576, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x16000) = 0x7fcad2347000
mmap(0x7fcad234d000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b000) = 0x7fcad234d000
mmap(0x7fcad234f000, 13424, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fcad234f000
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fcad232f000
arch_prctl(ARCH_SET_FS, 0x7fcad232fc00) = 0
mprotect(0x7fcad2523000, 16384, PROT_READ) = 0
mprotect(0x7fcad234d000, 4096, PROT_READ) = 0
mprotect(0x7fcad2357000, 4096, PROT_READ) = 0
mprotect(0x7fcad27ec000, 196608, PROT_READ) = 0
mprotect(0x55c8fea72000, 4096, PROT_READ) = 0
mprotect(0x7fcad2854000, 4096, PROT_READ) = 0
munmap(0x7fcad2824000, 21269)           = 0
set_tid_address(0x7fcad232fed0)         = 4088
set_robust_list(0x7fcad232fee0, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7fcad2337690, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7fcad2344140}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7fcad2337730, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7fcad2344140}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
readlink("/proc/self/exe", "/home/admin/IU-advanced-linux/la"..., 4096) = 44
getxattr("/home/admin/IU-advanced-linux/lab-2/hack_app", "user.license", 0x55c8fea73040, 4096) = -1 ENODATA (No data available)
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL)                               = 0x55c8ffa42000
brk(0x55c8ffa63000)                     = 0x55c8ffa63000
write(1, "Welcome to Lab2 super secure pro"..., 38Welcome to Lab2 super secure program!
) = 38
write(1, "Your HWID is A0060600FFFB8B17.\n", 31Your HWID is A0060600FFFB8B17.
) = 31
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "Enter the license key: ", 23Enter the license key: ) = 23
read(0, 0x55c8ffa426b0, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, 0x55c8ffa426b0, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, 0x55c8ffa426b0, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, 0x55c8ffa426b0, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, 0x55c8ffa426b0, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0,
```

As we can see there was some attempt to read saved key from file attributes, but it faults, because we have not such attribute.

So let’s take a look at result of `Ghidra` decompilation

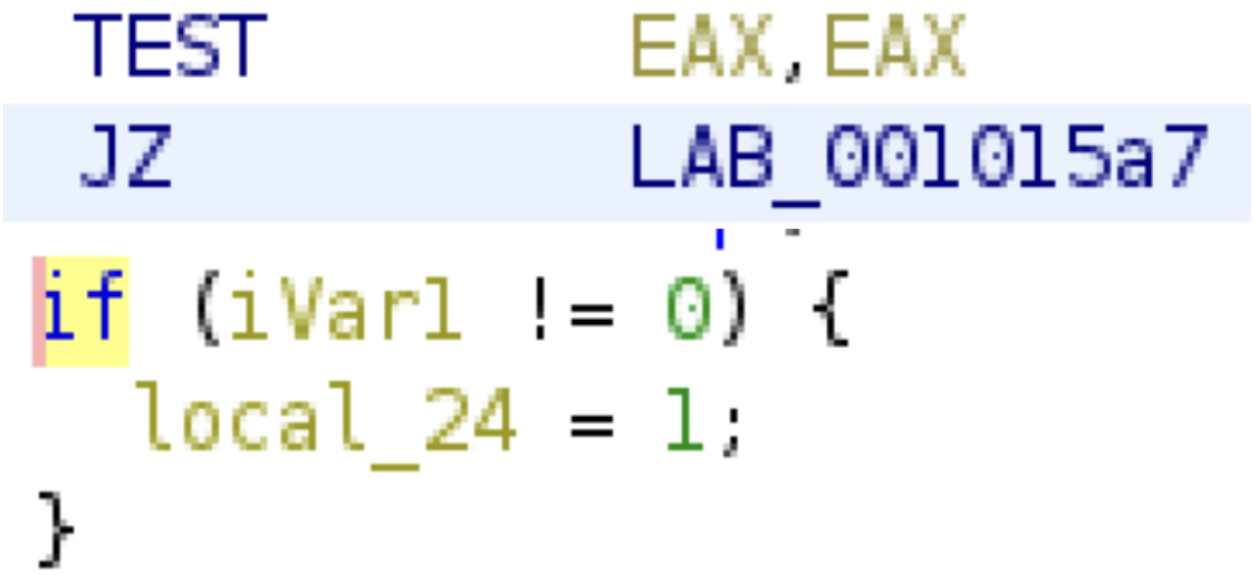
```
28  __get_cpuid(1,&local_34,&local_30,&local_2c,&local_28,in_R9,param_2);
29  local_18 = local_34 << 0x18 | local_34 >> 0x18 | (local_34 & 0xff00) << 8 | local_34 >> 8 & 0xff00
30  ;
31  local_14 = local_28 << 0x18 | local_28 >> 0x18 | (local_28 & 0xff00) << 8 | local_28 >> 8 & 0xff00
32  ;
33  snprintf(PSN,0x11,"%08X%08X",(ulong)local_18,(ulong)local_14);
34  calc_md5(PSN,0x10);
35  for (local_20 = 0; local_20 < 0x10; local_20 = local_20 + 1) {
36    sprintf(md5decode + local_20 * 2,"%02x",(ulong)(byte)md5digest[0xf - local_20]);
37  }
38  readlink("/proc/self/exe",binaryPath,0x1000);
39  getxattr(binaryPath,"user.license",xattrValue,0x1000);
40  puts("Welcome to Lab2 super secure program!");
41  iVar1 = strncmp(md5decode,xattrValue,0x21);
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: ",(ulong)local_18,(ulong)local_14);
47    __isoc99_scanf(&DAT_0010208f,userInput);
48    iVar1 = strncmp(md5decode,userInput,0x21);
49    if (iVar1 == 0) {
50      setxattr(binaryPath,"user.license",md5decode,0x21,0);
51      puts("Now you app is activated! Thanks for purchasing!");
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 */
63    __stack_chk_fail();
```

In line 41 we can see that here is the comparison of attribute `xattrValue` and precalculated key, so if we invert result of validation in line 42, we can pass this validation

```
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

                     LAB_001015a7                                    )
001015a7 83 7d e4 00     CMP        dword ptr [RBP + local_24],0x0
```

If we replace `JNZ` to opposite `JZ` we can avoid jump after false in if condition in line 42



So we can see that `Ghydra` change decompiled listing, let's try export program and try to run already patched version

```
admin@debian-vm:~/IU-advanced-linux/lab-2$ ./hack_app.patched
Welcome to Lab2 super secure program!
Your app is licensed to this PC!
Press Enter to continue...
```

Success!

Keygen

After long time googling I found implementation of __get_cpuid from GCC, which we can find in line 28, this method gives us information about the processor

```
28  __get_cpuid(1,&local_34,&local_30,&local_2c,&local_28,in_R9,param_2);
29  local_18 = local_34 << 0x18 | local_34 >> 0x18 | (local_34 & 0xff00) << 8 | local_34 >> 8 & 0xff00
30  ;
31  local_14 = local_28 << 0x18 | local_28 >> 0x18 | (local_28 & 0xff00) << 8 | local_28 >> 8 & 0xff00
32  ;
```

`local_34` is about processor version

`local_28` is about processor features

After much thought, I came to the conclusion that:

`local_18` is `bytewise_reverse(local_34)`

`local_14` is `bytewise_reverse(local_28)`

And we can figure out that

`key = hex(reverse(MD5(HWID)))`

```python
import hashlib

HWID = input("Enter your HWID: ")
result = hashlib.md5(HWID[0:0x10].encode())
print(result.digest()[::-1].hex())
```

Let’s try run keygen for my HWID

```
admin@debian-vm:~/IU-advanced-linux/lab-2$ ./hack_app
Welcome to Lab2 super secure program!
Your HWID is A0060600FFFB8B17.
Enter the license key: 123
Provided key is wrong! App is closing!
Press Enter to continue...
admin@debian-vm:~/IU-advanced-linux/lab-2$ python3 keygen.py
Enter your HWID: A0060600FFFB8B17.
dd3dc694de818b9ab7c834c72308ea8a
admin@debian-vm:~/IU-advanced-linux/lab-2$ ./hack_app
Welcome to Lab2 super secure program!
Your HWID is A0060600FFFB8B17.
Enter the license key: dd3dc694de818b9ab7c834c72308ea8a
Now you app is activated! Thanks for purchasing!
Press Enter to continue...
admin@debian-vm:~/IU-advanced-linux/lab-2$
```

As we can see it works

Patch

With python I found which byte I changed with `Ghydra` , so remains to change it

```python
with open('hack_app', 'rb') as f:
    data = f.read()

patched_data = data[:5534] + (116).to_bytes(1, 'little') + data[5534+1:]

with open('patched_hack_app', 'wb+') as f:
    f.write(patched_data)      You, 1 second ago • Uncommitted changes
```

```
admin@debian-vm:~/IU-advanced-linux/lab-2$ ./patched_hack_app
Welcome to Lab2 super secure program!
Your app is licensed to this PC!
Press Enter to continue...
```