

Reverse Engineering TTC6510-3002

Joonatan Ovaska

A K M MAHMUDUL HAQUE
AB0208

Student number: 2110841

Lab02

Date: 13.09.2023

Steps

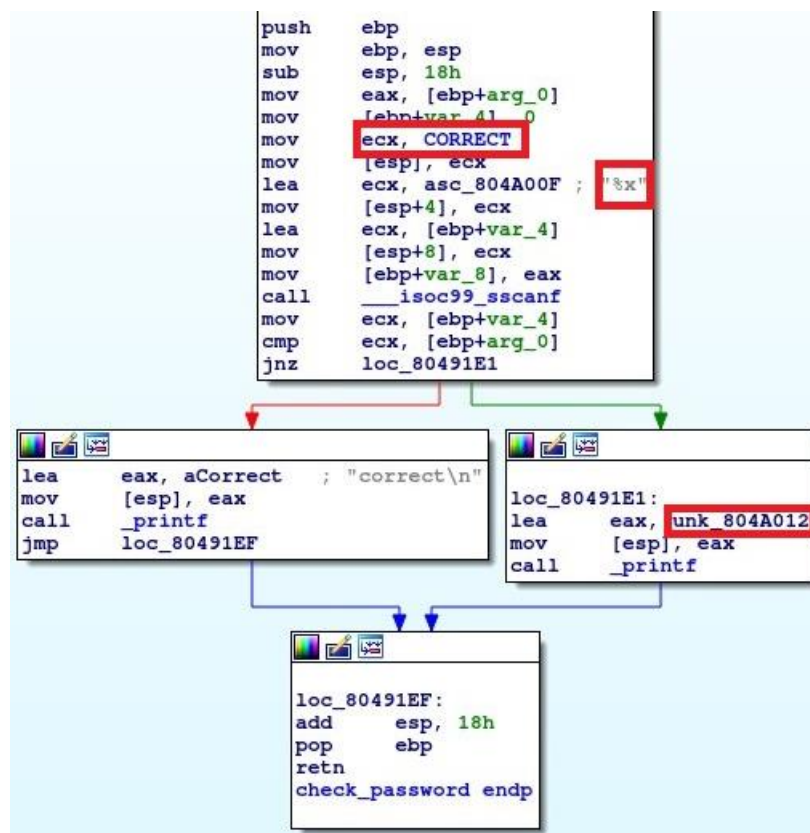
- At the **Main** function the comment “**Insert password**” grabbed my attention then reading thoroughly gives me confidence when in called for print.
- By **%d** my guess is that the input should be a decimal.

```

push    ebp
mov     ebp, esp
sub     esp, 28h
mov     eax, [ebp+argv]
mov     ecx, [ebp+argc]
mov     [ebp+var_4], 0
mov     [ebp+var_8], 0
lea     edx, aPassword ; "Password: "
mov     [esp], edx
mov     [ebp+var_C], eax
mov     [ebp+var_10], ecx
call    _printf
lea     ecx, aD ; "%d"
mov     [esp], ecx
lea     ecx, [ebp+var_8]
mov     [esp+4], ecx
mov     [ebp+var_14], eax
call    __isoc99_scanf

```

- Next, the action goes to the function call **check_password** where it needs digging.



- By **%x** it depicts that the input is converted in string value, so I should look for some string for my guess also.

- Then digging into **Correct** in the **ecx** registry opens up the following where I can see a string **OxBEEF** (it is also a hexadecimal value, a point of suspicion).

```

.data:0804C024 public CORRECT
.data:0804C024 CORRECT dd offset a0xbeef ; DATA XREF: check_password+10+r
.data:0804C024 _data ends ; "OxBEEF"
.data:0804C024
.bss:0804C028 ;
.bss:0804C028

```

- Later digging into **unk_804A012** also shows the same **OxBEEF** hexadecimal value that can be taken into consideration.

```

.rodata:0804A006 db 2
.rodata:0804A007 db 0
.rodata:0804A008 a0xbeef db OxBEEF' 0 ; DATA XREF: .data:CORRECT+0
.rodata:0804A00F asc_804A00F db 'x',0 ; DATA XREF: check_password+19+o
.rodata:0804A012 unk_804A012 db 69h ; i ; DATA XREF: check_password:loc_80491E1+o
.rodata:0804A013 db 6Eh ; n
.rodata:0804A014 aCorrect db 'correct',0Ah,0 ; DATA XREF: check_password+3E+o
.rodata:0804A01D aPassword db 'Password: ',0 ; DATA XREF: main+1A+o
.rodata:0804A028 aD db '%d',0 ; DATA XREF: main+2E+o
.rodata:0804A028 _rodata ends

```

- Converting the hexadecimal **OxBEEF** value into decimal we get **48879**, which later input as the password.

```

(kali@kali-vle)-[~/Documents/Labs01/ReverseEngineeringLinuxLabs]
$ ./lab02-ver2
Password: 48879
correct

```

- The above input proves the password.

Topic	Time
Lab02	2 hours
Report writing	1 hours