Software Security- Assignment3 – Binary Analysis

Debuggers

The binary has an anti-debugger trigger, which stops the execution of the program if a debugger is detected. The anti-debugger trigger is done by calling the ptrace-debuggin functionality in the code. Because the debugger is already in use the function call will fail and the program execution is stopped.

This can be seen easile when running strace on the binary, which will have the following line in its output:

ptrace(PTRACE_TRACEME, 0, 0x1, 0) = -1 EPERM (Operation not permitted)

When looking at the objdump of the programm we can see where the ptrace-call is made in the binary.

400dda: e8 a1 fc ff ff callq 400a80 <ptrace@plt>

We replace this line in the binary with NOPs (0x90), after which the ptrace is not called anymore and the binary can be also run with a debugger.

Time-based Trigger

When the binary is run, it gives the following output: "'arlogh Qoylu'pu'?".

This is a phrase in Klingon language and means “What time is it?”.

Using ltrace we can see that the binare calls the function: gettimofday. When we debug the spot in the binary where the call is made we can see the following happening:

```
400ea9:   e8 32 fc ff ff          call   400ae0 <gettimeofday@plt>
400eae:   bf 1e 00 00 00          mov    edi,0x1e
400eb3:   48 8b 75 f0             mov    rsi,QWORD PTR [rbp-0x10]
400eb7:   48 b9 00 00 ff ff 00    movabs rcx,0xffff0000
400ebe:   00 00 00
400ec1:   48 21 ce                and    rsi,rcx
400ec4:   48 81 ce de c0 00 00    or     rsi,0xc0de
400ecb:   48 89 34 25 d0 27 40    mov    QWORD PTR ds:0x4027d0,rsi
```

Basically the time is read to the rsi-register and then modified in a way, that for example out of 0x58881234 becomes 0x5888c0de, which is then stored to 0x4027d0. This location of the memory is then checked in a function(starting at 0x400fa0) that is called in beginning of main. After investigating the flow of that function I was able to detect the following criterias (lets say if the time-stored was 0x5888c0de, we call it 0xABCD):

*# (B*C+A) & 0xff == 0x2f*

 # A^D^(B+C) & 0xff == 0x5b

I then created a python program (attached) to calculate the values, for the time, when we assume that the C and D (0xc0de) are unchanged. The values that pass these criterias are:

A=0xaf
B=0x6a
Using the debugger we then modify the timevalue being stored to 0x4027d0 to be the following:

0xaf6ac0de

When this is done, the program output changes to:

This is not the secret you are looking for!
To dig deeper you have!

Secret Message

When the memory is investigated at the location where the suggestion to "dig deeper" is printed, we can see that the actual data at the string location is:

x /s 0x404000
0x404000: "\353n\nThe secret is: \"The truth is out there!\"\nThis is not the secret you are looking for!\nTo dig deeper you have!\nH1\300H1\377H1\322H\203\300\001H\203\307\001H\215\065\244\377\377\377H\203\302D\017\005H1\300H\203\300<H1\377\017\005"

According to this text the secret message is: "The truth is out there!".