Project 2, part 2: Password

By Chris Lin

Lin9\_1:

0x08048e7f <+47>: call 0x804f550 <fgets>

0x08048e87 <+55>: call 0x8048e24 <chomp>

0x08048e96 <+70>: repz cmpsb %es:(%edi),%ds:(%esi)

0x08048eb9 <+105>: call 0x806ef00 <\_\_printf\_chk>

0x08048ec7 <+119>: call 0x804f9a0 <puts>

0x08048ede <+142>: call 0x806f030 <\_\_stack\_chk\_fail>

Passphrase: VWAIgKvTHiIoaexTPwTC

In the first compiled executable, I set the first break point at main as usual, then I disassemble the code to get an idea of what the program is doing. I noted all the places with function call and a specific line that seems important to me, 0x08048e96 <+70>: repz cmpsb %es:(%edi),%ds:(%esi). It is comparing the string I input and with the correct passphrase. So, I assumed the correct passphrase will be in register %esi. I found the correct passphrase as VWAIgKvTHiIoaexTPwTC.

Lin9\_2:

0x08048615 <+50>: call 0x80483b0 <fgets@plt>

0x0804861d <+58>: call 0x804852c <c>

0x08048627 <+68>: call 0x8048555 <x>

0x08048633 <+80>: call 0x8048598 <e>

0x08048653 <+112>: call 0x8048400 <\_\_printf\_chk@plt>

0x08048661 <+126>: call 0x80483d0 <puts@plt>

0x08048678 <+149>: call 0x80483c0 <\_\_stack\_chk\_fail@plt>

Pass: lin9\_2

In the second compiled executable, I also set the first breakpoint at main and disassemble the code. I am interested by

0x08048633 <+80>: call 0x8048598 <e> since it is the last call that do something to the string and followed with test underneath. I am thinking that the test is comparing string in some way. I set my second breakpoint at 0x08048633, then I exanimate eax register which is compare by test. I found the passphrase is lin9\_2.

Lin9\_3:

Use mystrings to find the label of main:

libc.so.6

\_IO\_stdin\_used

puts

\_\_stack\_chk\_fail

printf

getchar

\_\_libc\_start\_main

\_\_gmon\_start\_\_

GLIBC\_2.4

GLIBC\_2.0

Disas:

0xf7e0fa04 <+4>: call 0xf7f1ecbb

0xf7e0fa23 <+35>: test %edx,%edx

0xf7e0fa3a <+58>: test %eax,%eax

0xf7e0fa69 <+105>: test %esi,%esi

0xf7e0fa53 <+83>: call 0xf7e29490 <\_\_cxa\_atexit>

0xf7e0fa88 <+136>: call \*%esi

0xf7e0faad <+173>: call 0xf7e24450 <\_setjmp>

0xf7e0fab2 <+178>: test %eax,%eax

0xf7e0faef <+239>: call \*0x70(%esp)

0xf7e0faf6 <+246>: call 0xf7e29260 <exit>

0xf7e0fb12 <+274>: call \*%eax

0xf7e0fb2c <+300>: test %dl,%dl

0xf7e0fb37 <+311>: call 0xf7ed23b0

0xf7e0fb55 <+341>: call \*0x1a4(%eax)

0xf7e0fb81 <+385>: test %eax,%eax

0xf7e0fb8c <+396>: call \*%eax

ç <+411>: cmp %ebp,%edi

0xf7e0fbbb <+443>: call \*0x1a4(%eax)

Passphrase:

In the last compiled executable, I set a break point on main but saying it’s not defined. Then, I run lin9\_3 with the mystrings program and find the correct label for main, which is \_\_libc\_start\_main. I was able to set my first breakpoint on the main label I found. After that, I found I couldn’t access many of the memories, and setting the next breakpoint becomes a problem. The second breakpoint I set at the end is getchar getting from mystrings output. I was stuck on seeing the output from the register and was not able to find the correct passphrase.