Assignment 1

CS59

October 7, 2024

This assigment will help make you more proficient with the Arm assembler and debugging.

You have been given a program called "**f00xxxx_a1**", where "xxxx" is replaced by your student number. It should be in your home directory under

CS59/VirtualMachine/VM_Shared_Directory

on *thepond*. This will allow you to access the file through the QEMU virtual machine under the **/mnt** directory. If you didn't get an executable, let Pete Brady know as soon as possible.

To start the virtual machine, run the **launch_vm.sh** script in **CS59/VirtualMachine**. This will start the Alpine Linux session in your terminal window. The login is *root* with no password. To shut down the virtual machine when you're done, use the **poweroff** command.

Your program is very straightforward; execute the program, enter the correct code on the command line and a token will be returned to you. If you type in an incorrect code, the program exits. The code is an encrypted version of the plaintext "hello_world".

For example, we'll use a sample program that is similar to yours (each program has slightly different variables), so you can either type the encryption code after the program starts:

```
aarch64-alpine-vw:~# cd /mnt
aarch64-alpine-vw:/mnt# ./f002w4q_a1
Enter the input: hello_world
Incorrect input.
```

Or you can echo the input into the program:

```
aarch64-alpine-vw:/mnt# echo "hello_world" | ./f002w4q_a1
Enter the input: Incorrect input.
```

Once you use the debugger to figure out how the program works, you can enter the correct input to get a token out:

```
aarch64-alpine-vw:/mnt# ./f002w4q_a1
Enter the input: <input>
Your token is: <token>
aarch64-alpine-vw:/mnt#
```

Use the **lldb** debugger to explore how the encryption algorithm works, which will help you solve the problem. Once you figure that out, you should be able to enter the encrypted value and get your token. For example:

```
aarch64-alpine-vw:/mnt# lldb ./f002w4q_a1
(lldb) target create "./f002w4q_a1"
Current executable set to '/mnt/f002w4q_a1' (aarch64).
(lldb) b main
Breakpoint 1: where = f002w4q_a1`main, address = 0x0000000000000a08
(lldb) r
Process 2477 launched: '/mnt/f002w4q_a1' (aarch64)
Process 2477 stopped
* thread #1, name = 'f002w4q_a1', stop reason = breakpoint 1.1
    frame #0: 0x0000aaaaaaaa0a08 f002w4q_a1`main
```

```
f002w4q_a1`main:
->  0xaaaaaaaaa0a08 <+0>:  sub    sp, sp, #0xc0
    0xaaaaaaaaa0a0c <+4>:  stp    x29, x30, [sp, #0xa0]
    0xaaaaaaaaa0a10 <+8>:  add    x29, sp, #0xa0
    0xaaaaaaaaa0a14 <+12>: stp    x19, x20, [sp, #0xb0]
(lldb) dis
f002w4q_a1`main:
->  0xaaaaaaaaa0a08 <+0>:   sub    sp, sp, #0xc0
    0xaaaaaaaaa0a0c <+4>:   stp    x29, x30, [sp, #0xa0]
    0xaaaaaaaaa0a10 <+8>:   add    x29, sp, #0xa0
    0xaaaaaaaaa0a14 <+12>:  stp    x19, x20, [sp, #0xb0]
    0xaaaaaaaaa0a18 <+16>:  adrp   x0, 31
    0xaaaaaaaaa0a1c <+20>:  ldr    x0, [x0, #0xfb8]
```

This shows loading the program with lldb, setting a breakpoint at main, running the program, getting a breakpoint, then printing out the disassembled code at that point.

Completing the Assignment

You will need to turn in the following:

1. Explain the encryption method that was used in your program.
2. Explain the method you used to arrive at that conclusion. There are several different methods you can use to solve this problem.
3. Provide the encrypted token that the program returned and the plaintext of that token.

The assignment is due by Monday October 14th.