**Diep (Emma) Vu**

# Lab 8 Report

**The password is:** `swrdfish`
**The way I found the password:**

First, I converted the `.hex` file to MIPS assembly instructions by using the online disassembler. Here are the tools I used:
https://blog.loadzero.com/blog/announcing-mipsdis/ for converting the instructions with the correct address (but the registers are numbers and I figure it is easier to see them by name)
https://www.eg.bucknell.edu/~csci320/mips_web/ for converting the number registers to name registers (this tool doesn't give out the correct address)

Then, I looked for the `jal, j` instructions to an address and found out here are the addresses that I need to pay attention to:

```
#0x4000a0 #label2
#0x40003c #loop
#0x400024 #label1
#0x4000f0 #label3
#0x40010c #branch1
```

I picked up the hints from John Daly in Slack that `lui` followed by `ori` instructions take the most 2 significant and least significant bytes in order to get the address. For example,

```
lui $at, 0x1001($0)
ori $s3, $at, 0x90 #txsegjti
```

Register `$s3` will be stored in the address `0x10010090` so I look at the `passwdmemdump.txt` file and found out the value at that address is `txsegjti`

It is tricky because there are a lot of traps in hinting at the password like `honeypot` when I look at `$s1`. This line also gave me a hint that password has at most 8 characters because register `$v0` is the return value: `addi $v0, $0, 0x8`

Jing also gave me hints that I need to look for the instructions that compare the user input and the password and pay attention to the address that the jump instructions lead to.

I typed out comments line by line to trace the changes of `$a0` and `$a1` because that's the user input and actual password. There's a code block that compares the

user input and actual password character by character by using a loop. Here are the 4 most important lines:

```
lb $t1, 0x0($a0) #t1 = each character in txsegjti
lb $t2, 0x0($a1) #
addi $t2, $t2, 0x1 #t2 = t2 + 1
bne $t2, $t1, 0x4 #compare t1 to t2+1
```

I realized that the instruction `addi $t2, $t2, 0x1` shifts all the characters by 1 and the code is comparing whether `$t1` matches `$t2 + 1` so that means `txsegjti` is the decoded message to compare for matching and in order to find the actual password, we have to shift each character in this message left by 1. So I wrote out the alphabet on paper and figure out that `txsegjti` with all characters left shift by 1 is `swrdfish`

This is a very fun lab and definitely the thing I would remember the most after this class. I found it helpful to comment on the code line by line to trace the address of these registers and definitely the hints from Jing and John Daly's message in Slack helped. I literally cried after figuring out because I felt so happy yet dumb.

Apparently, Jing told me Matt accidentally slipped out the password during CS Fest (which makes sense since he's the villain). However, I could never recall hearing anything about `swrdfish` at CS Fest because all I remembered was Murder Mystery, Matt killing Valerie to prevent ChatGPT.

I may or may not change all of my Union passwords to this one (*wink wink)

**Pop culture reference:** `swrdfish` is the name of the movie in 2001 starring by Halle Berry, and Hugh Jackman. However, this movie had really bad ratings so I don't know if I want to watch it.