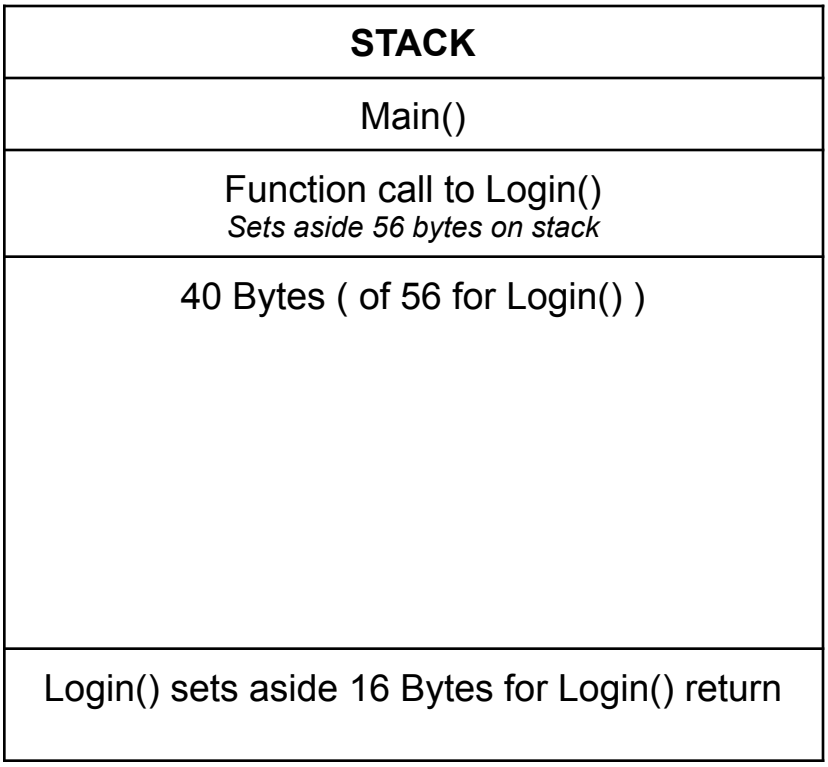
Melanie Prettyman
03/22/24
CS6014



Login(), sets aside 56 bytes on the stack. At rsp + 16, login() sets aside 16 bytes on the stack to the address of the return call (results to failure() or success() functions). That leaves the other 40 bytes to store the password. By overwriting the 40 bytes of the password and then tacking on the address for the success call, we can bypass the actual password comparison and alway jump to the success call in Login.
This was done by padding 40 "a"s to the password text file followed by the address of success in hexadecimal in little endian.

To find the actual address of success I ran the c file through lldb debugger.

```
➜  BufferOverFlowLab git:(main) ✗ lldb ./a.out
(lldb) target create "./a.out"
Current executable set to
'/Users/melanieprettyman/desktop/msd/spring2024/6014/cs6014/BufferOverFlowLab/a.
out' (x86_64).
(lldb) run
Process 627 launched:
'/Users/melanieprettyman/desktop/msd/spring2024/6014/cs6014/BufferOverFlowLab/a.
out' (x86_64)
```

```
warning: libobjc.A.dylib is being read from process memory. This indicates that LLDB
could not read from the host's in-memory shared cache. This will likely reduce
debugging performance.
enter your password:
Process 627 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS
(code=1, address=0x0)
    frame #0: 0x0000000100003de0
->  0x100003de0: addb   %al, (%rax)
    0x100003de2: addb   %al, (%rax)
    0x100003de4: addb   %al, (%rax)
    0x100003de6: addb   %al, (%rax)
Target 0: (a.out) stopped.
(lldb) p success
(void (*)()) 0x0000000100000de0 (a.out`success at login.c:21)
```

Here I obtained the actual address of success in the stack.