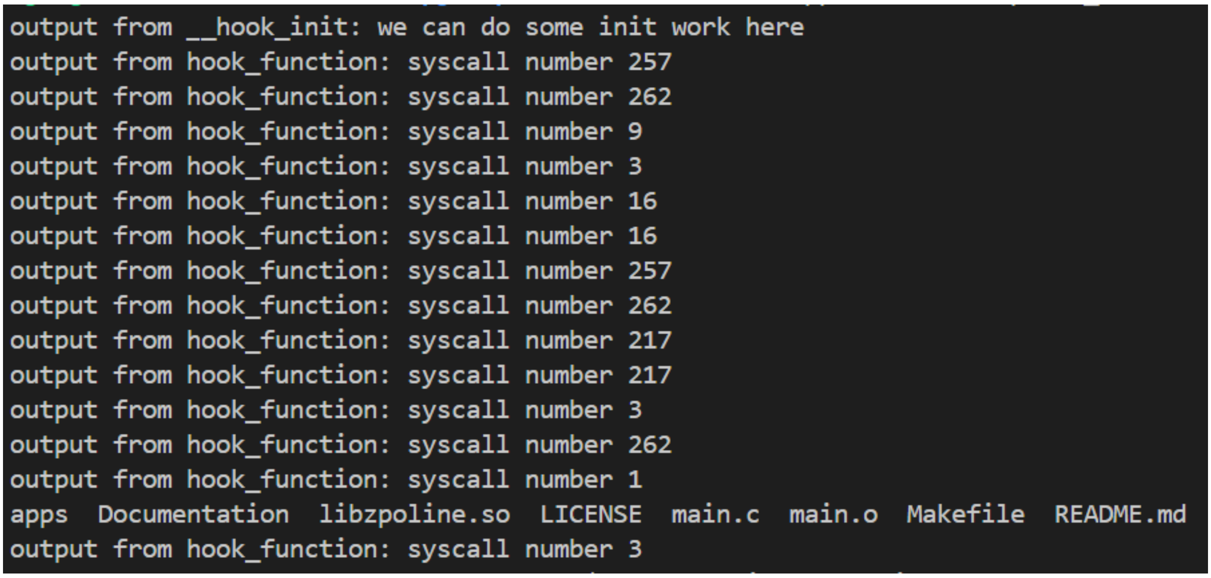
Project 1 Report

110550158 葉家蓁

Task1:

(a) Output



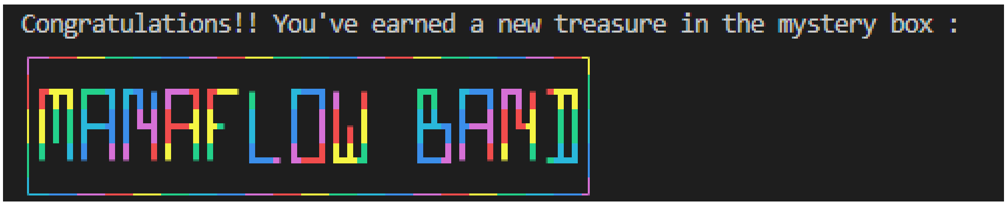
```
output from __hook_init: we can do some init work here
output from hook_function: syscall number 257
output from hook_function: syscall number 262
output from hook_function: syscall number 9
output from hook_function: syscall number 3
output from hook_function: syscall number 16
output from hook_function: syscall number 16
output from hook_function: syscall number 257
output from hook_function: syscall number 262
output from hook_function: syscall number 217
output from hook_function: syscall number 217
output from hook_function: syscall number 3
output from hook_function: syscall number 262
output from hook_function: syscall number 1
apps  Documentation  libzpoline.so  LICENSE  main.c  main.o  Makefile  README.md
output from hook_function: syscall number 3
```

(b) Find System Call

257	sys_openat
262	sys_newfstatat
9	sys_mmap
3	sys_close
16	sys_ioctl
16	sys_ioctl
257	sys_openat
262	sys_newfstatat
217	sys_getdents64
217	sys_getdents64
3	sys_close
262	sys_newfstatat
1	sys_write
3	sys_close

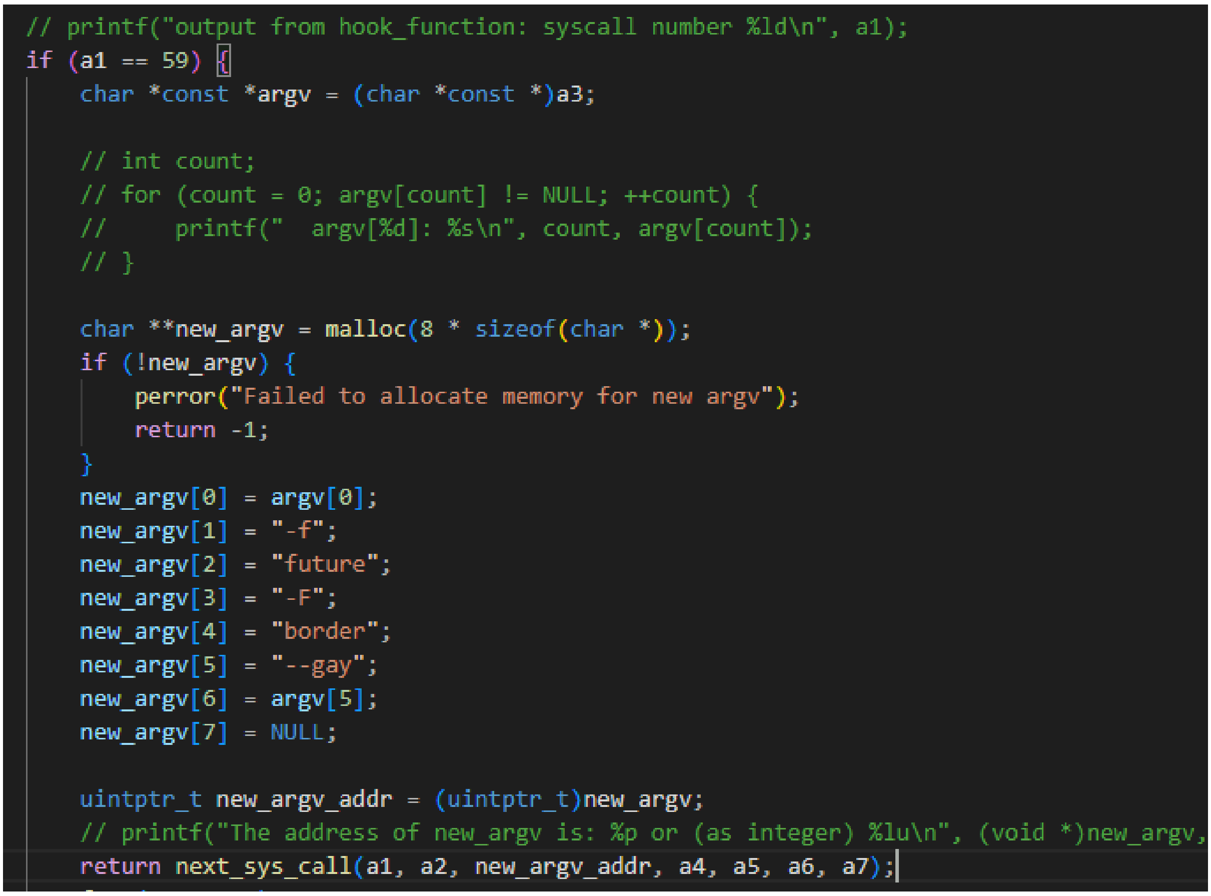
Task2:

(a)



```
Congratulations!! You've earned a new treasure in the mystery box :
```

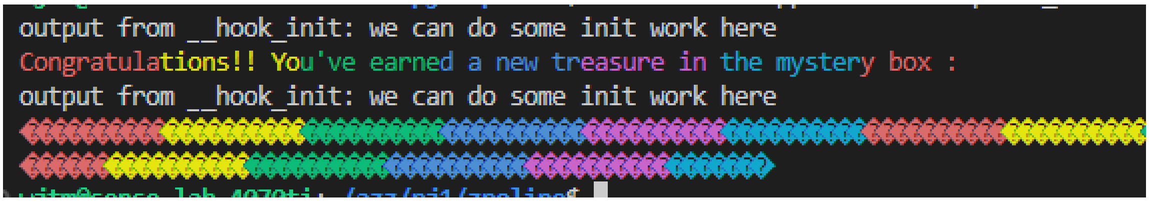
(b) I found that syscall 59 is a toilet command and I use a new argv to change the design of the output.



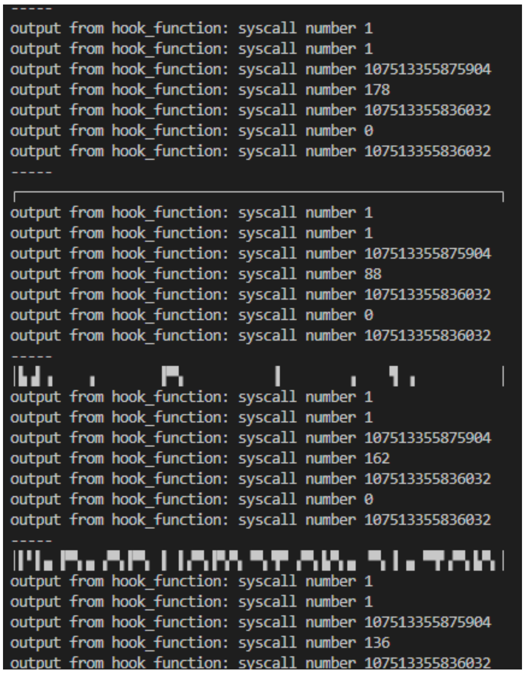
```
// printf("output from hook_function: syscall number %ld\n", a1);
if (a1 == 59) {
    char *const *argv = (char *const *)a3;

    // int count;
    // for (count = 0; argv[count] != NULL; ++count) {
    //     printf("  argv[%d]: %s\n", count, argv[count]);
    // }

    char **new_argv = malloc(8 * sizeof(char *));
    if (!new_argv) {
        perror("Failed to allocate memory for new argv");
        return -1;
    }
    new_argv[0] = argv[0];
    new_argv[1] = "-f";
    new_argv[2] = "future";
    new_argv[3] = "-F";
    new_argv[4] = "border";
    new_argv[5] = "--gay";
    new_argv[6] = argv[5];
    new_argv[7] = NULL;

    uintptr_t new_argv_addr = (uintptr_t)new_argv;
    // printf("The address of new_argv is: %p or (as integer) %lu\n", (void *)new_argv,
    return next_sys_call(a1, a2, new_argv_addr, a4, a5, a6, a7);
```

Question:
I suppose that is not a feasible solution. It might cause security problems since vDSO is provided by the kernel to user space. It maps a small piece of kernel code into virtual address space of user space. Modifying the kernel code might introduce security vulnerabilities. In addition, it might also cause system stability problems. The system might crash if it performs a binary patch.

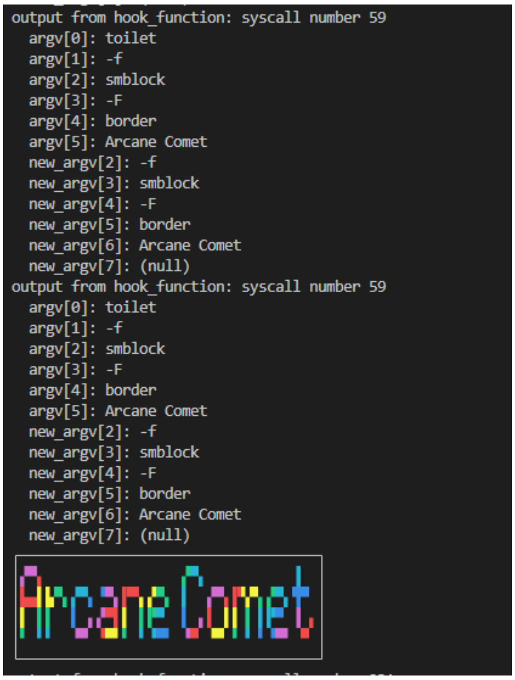
Thoughts:
This homework seems simple at first. Then I was stuck for a long time because I didn't even know where to start. I also have zero experience in reverse engineering. I tried many ways and even printed out things like the pictures below. When I tried to add colors on the ascii art and I totally went the wrong way. I thought I should add color when syscall 1, but actually I only need to modify the toilet command when syscall 59 and do execve() with new argv.



```
output from __hook_init: we can do some init work here
Congratulations!! You've earned a new treasure in the mystery box :
output from __hook_init: we can do some init work here
yitm@sense-lab-4070ti: /ezz/pj1/zpoline$
```



```
-----
output from hook_function: syscall number 1
output from hook_function: syscall number 1
output from hook_function: syscall number 107513355875904
output from hook_function: syscall number 178
output from hook_function: syscall number 107513355836032
output from hook_function: syscall number 0
output from hook_function: syscall number 107513355836032
-----

output from hook_function: syscall number 1
output from hook_function: syscall number 1
output from hook_function: syscall number 107513355875904
output from hook_function: syscall number 88
output from hook_function: syscall number 107513355836032
output from hook_function: syscall number 0
output from hook_function: syscall number 107513355836032
-----

output from hook_function: syscall number 1
output from hook_function: syscall number 1
output from hook_function: syscall number 107513355875904
output from hook_function: syscall number 162
output from hook_function: syscall number 107513355836032
output from hook_function: syscall number 0
output from hook_function: syscall number 107513355836032
-----

output from hook_function: syscall number 1
output from hook_function: syscall number 1
output from hook_function: syscall number 107513355875904
output from hook_function: syscall number 136
output from hook_function: syscall number 107513355836032
```



```
output from hook_function: syscall number 59
  argv[0]: toilet
  argv[1]: -f
  argv[2]: smblock
  argv[3]: -F
  argv[4]: border
  argv[5]: Arcane Comet
  new_argv[2]: -f
  new_argv[3]: smblock
  new_argv[4]: -F
  new_argv[5]: border
  new_argv[6]: Arcane Comet
  new_argv[7]: (null)
output from hook_function: syscall number 59
  argv[0]: toilet
  argv[1]: -f
  argv[2]: smblock
  argv[3]: -F
  argv[4]: border
  argv[5]: Arcane Comet
  new_argv[2]: -f
  new_argv[3]: smblock
  new_argv[4]: -F
  new_argv[5]: border
  new_argv[6]: Arcane Comet
  new_argv[7]: (null)
```