PART A

We need to implement the code in Part A so that the example `open()` hook works. Please check the TODO comments in “rootkit.c”.

PART B

1- We need a new hook for the `execve` syscall using the framework code from Part A.

The hook should print the name of all files being executed, and the effective UID of the user executing the file to syslog using `printk`. Example output:

```
Jan 28 20:49:17 USER2 kernel: [81423.749198] Executing /usr/bin/tail

Jan 28 20:49:17 USER2 kernel: [81423.749200] Effective UID 0

Jan 28 20:49:19 USER2 kernel: [81425.950497] Executing /bin/ls

Jan 28 20:49:19 USER2 kernel: [81425.950499] Effective UID 1000
```

2- We need to modify the hook code so that when the effective UID of the user executing an executable is equal to the value of the `root_uid` parameter, they are given uid/euid 0 (i.e. root privs). The `root_uid` parameter must be provided via the `insmod` command in `insert.sh` like the sys_call_table address, and not hard coded. Note that the `root_uid` parameter should be set to **our user's UID** to get root, not root's UID. We need to add this behaviour.

PART C

1- We need to implement a hook for the `getdents` system call which should print the name of all directory entries returned by a call to `getdents()` to syslog using `printk`. Sample output:

```
Oct  1 11:44:36 USER2 kernel: [ 2266.441674] getdents() hook invoked.

Oct  1 11:44:36 USER2 kernel: [ 2266.441704] entry: rootkit.o

Oct  1 11:44:36 USER2 kernel: [ 2266.441706] entry: .rootkit.mod.o.cmd

Oct  1 11:44:36 USER2 kernel: [ 2266.441708] entry: ..

Oct  1 11:44:36 USER2 kernel: [ 2266.441710] entry: insert.sh

Oct  1 11:44:36 USER2 kernel: [ 2266.441711] entry: rootkit.c

Oct  1 11:44:36 USER2 kernel: [ 2266.441712] entry: rootkit.mod.c

Oct  1 11:44:36 USER2 kernel: [ 2266.441714] entry: rootkit.ko
```

2- Modify the hook such that the `struct linux_dirent*` buffer we return to the calling process does not include any dirent's for filenames that start with `magic_prefix`.

The `magic_prefix` character array should be provided as a kernel module parameter given to `insmod` in the `insert.sh` script. We need to implement this parameter. Example output (from normal user term):

```
USER@USER2:/code/rootkit_framework/test$ touch \$sys\$_lol_hidden.txt
USER@CUSER2:/code/rootkit_framework/test$ ls -la
total 8
-rw-rw-r-- 1 USER USER 0 Oct  1 11:59 bar.txt
-rw-rw-r-- 1 USER USER 0 Oct  1 11:59 baz.txt
-rw-rw-r-- 1 USER USER 0 Oct  1 11:59 foo.txt
-rw-rw-r-- 1 USER USER 0 Oct  1 12:00 $sys$_lol_hidden.txt
USER@USER2:/code/rootkit_framework/test$ ls -la
total 8
drwxrwxr-x 2 USER USER    4096 Oct  1 12:00 .
drwxrwxr-x 5 USER USER    4096 Oct  1 11:59 ..
-rw-rw-r-- 1 USER USER    0 Oct  1 11:59 bar.txt
-rw-rw-r-- 1 USER USER    0 Oct  1 11:59 baz.txt
-rw-rw-r-- 1 USER USER    0 Oct  1 11:59 foo.txt
```