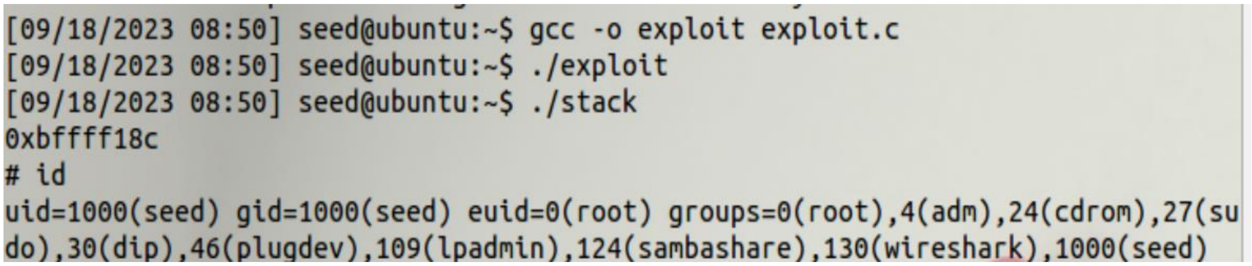
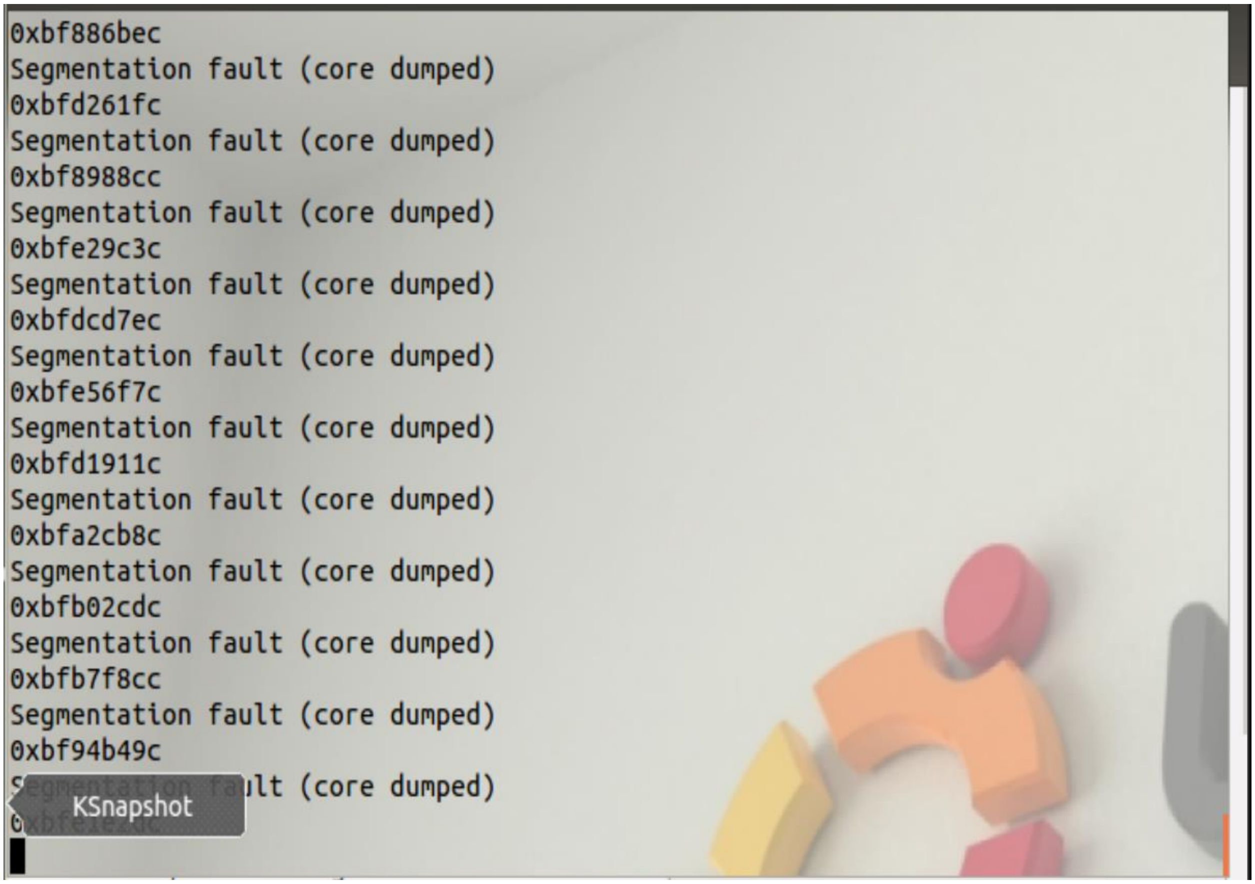
Austin Strobel

Task 1:



```
[09/18/2023 08:50] seed@ubuntu:~$ gcc -o exploit exploit.c
[09/18/2023 08:50] seed@ubuntu:~$ ./exploit
[09/18/2023 08:50] seed@ubuntu:~$ ./stack
0xbffff18c
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=0(root),4(adm),24(cdrom),27(su
do),30(dip),46(plugdev),109(lpadmin),124(sambashare),130(wireshark),1000(seed)
```

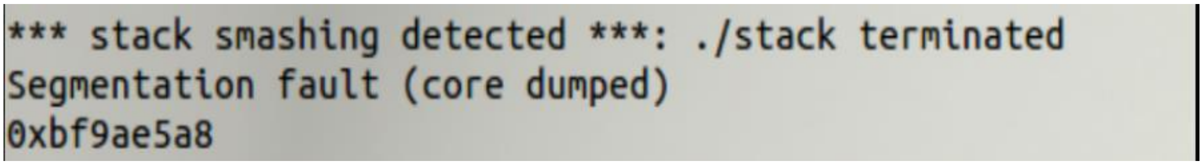
Task 2:



```
0xbf886bec
Segmentation fault (core dumped)
0xbfd261fc
Segmentation fault (core dumped)
0xbf8988cc
Segmentation fault (core dumped)
0xbfe29c3c
Segmentation fault (core dumped)
0xbfdcd7ec
Segmentation fault (core dumped)
0xbfe56f7c
Segmentation fault (core dumped)
0xbfd1911c
Segmentation fault (core dumped)
0xbfa2cb8c
Segmentation fault (core dumped)
0xbfb02cdc
Segmentation fault (core dumped)
0xbfb7f8cc
Segmentation fault (core dumped)
0xbf94b49c
Segmentation fault (core dumped)
0xbfe1e2dc
```

Due to ASLR (address space layout randomization), we see that there are random addresses output to the screen, but, eventually, the root address is output to the screen. These random addresses lead to continuous segmentation faults. This program attempts to guess various memory addresses until the root is found.

Task 3:



```
*** stack smashing detected ***: ./stack terminated
Segmentation fault (core dumped)
0xbf9ae5a8
```

The attacker tries to write more data into the canary than it can hold, creating a buffer overflow. If the canary has been modified, a buffer overflow has occurred. A segmentation fault occurs when the canary is modified and the program detects the modification. Stack Guard is enabled for this task. The stack smashing detection message means that the stack canary implementation is working as intended.