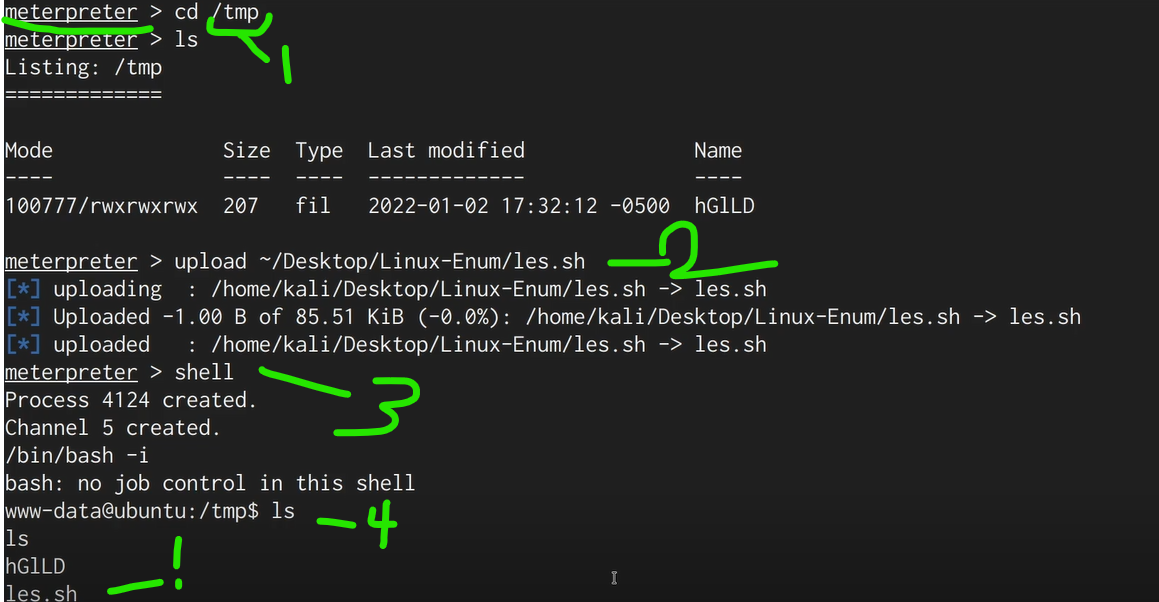
Starting scenario: we got a meterpreter session on a target system, but it’s some kind of low-level user that doesn’t have access to root privileges and cannot invoke sudo.

Escalate privileges, using something from exploit DB.

Move to the tmp file, or create one if it’s not there. We can upload the les.sh file to help us enumerate. This does some witchcraft that will help us later on.

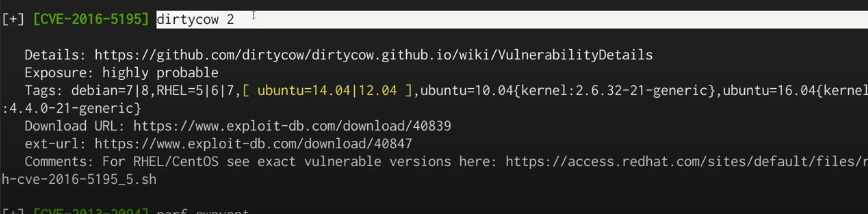
4) shows we got the file uploaded on the target system.



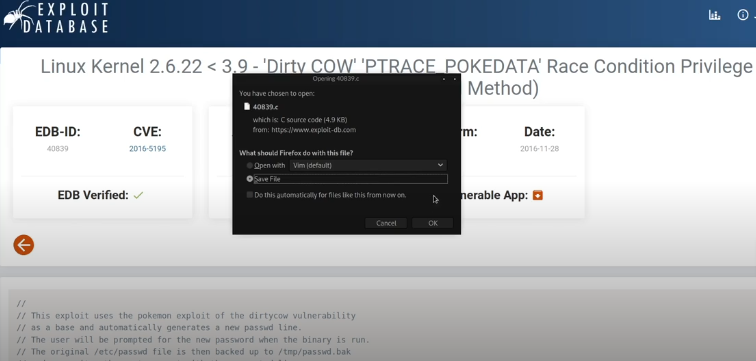
Running this enumeration gave us the following intel : target is ubuntu 14.04\12.04.

We can search exploit database for that.

There’s a special Cve called dirtycow 2 that will work on the target system, ubuntu 14.04.

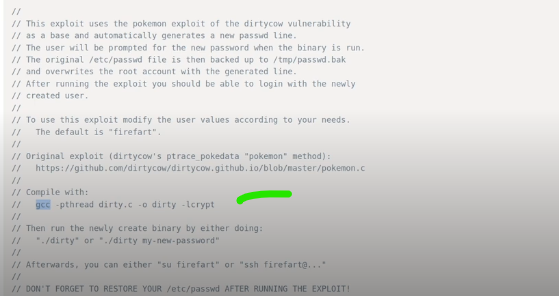


Exploit database will allow us to get the code for this

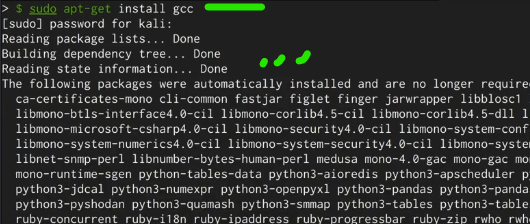


We should look at the code before doing anything. We can tell this is in C, the programming language, and it needs gcc to compile. This is deep-lore, but gcc us used to compile C.

Kinda weird that the default name is “firefart” borders on the scatological… then again, the bloody name of the program is “dirty cow”, although “COW” is the copy-on-write functionality that’s getting low-jacked here.



Install the gcc compiler if you don’t already have it. Again this is needed to compile C code, which dirty cow is written in



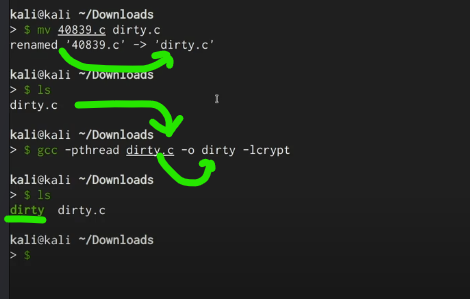
…

Change the download name from the random numbers to dirty.c

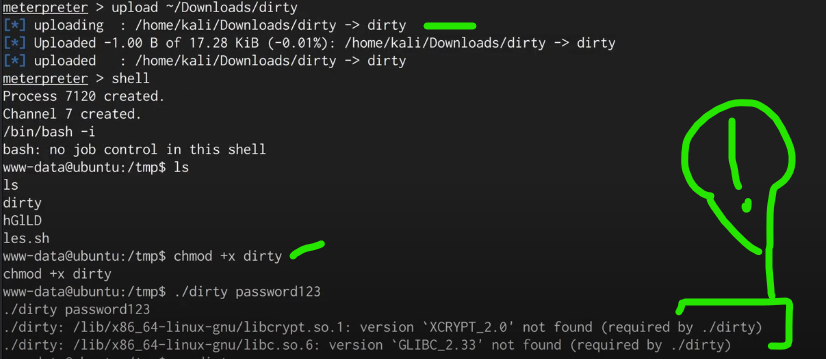
Compile it with gcc, make the output the filename “dirty”

(we found this compiling info within the comments of the exploit DB documentation)

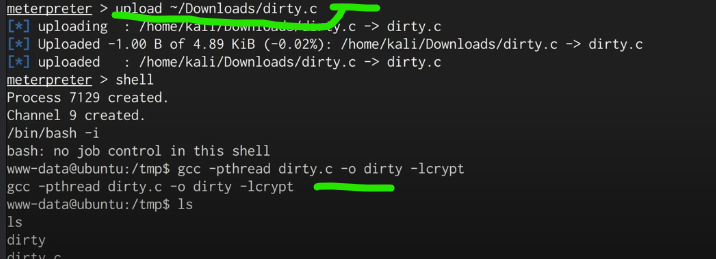
Now we got the compiled executable called ‘dirty’.



We try to upload it, and we give ourselves execute permissions, but the execution runs into a problem



We have to do the compilation on the target system through meterpreter… so, take the c file, get the shell running, then use the compiler on the file to make the executable on the target system.



This then can be executed by calling “dirty” in the meterpreter session, and we basically get root from it’s execution.