# ./Cow-tastrophy

The **Dirty COW** (*Dirty copy-on-write*, CVE-2016-5195) vulnerability is a notable **security flaw** in the **Linux kernel**, impacting all Linux-based systems prior to 2018.

It's a **local privilege escalation bug** that takes advantage of a race condition in the **copy-on-write** mechanism of the kernel's memory-management system. This vulnerability allows a local attacker to manipulate the **copy-on-write** mechanism to convert a **read-only** file mapping into a **writable one**, enabling them to write to memory mappings that should be read-only. Successfully exploited, it can grant an attacker elevated privileges and **root** access on the system.

Steps to Exploit Dirty COW:

- **Target Process Identification**: Find a process on the system suitable for the attack.
- **Race Condition Creation**: Continuously write to a targeted memory page while the kernel marks it as read-only.
- **Malicious Code Injection**: Use ptrace with **PTRACE_POKEDATA** to write malicious code into the targeted page.
- **Race Condition Exploitation**: Due to the race condition, the kernel might update the page's permissions based on outdated information, allowing the attacker's modifications to persist.

In Our Case:

- **Compiling**: Compile using **gcc -pthread dirty.c -o dirty -lcrypt** and execute with **./dirty [password]**.
- **File Backup**: The exploit starts by backing up **/etc/passwd** to **/tmp/passwd.bak**.
- **User Information Setup**: A user structure is defined and set up with **root** privileges.
- **Password Hashing and Line Generation**: It hashes a password and creates a formatted password line for **/etc/passwd**.
- **Memory Mapping**: Maps **/etc/passwd** into memory for modification.
- **Fork and Race Condition**: The exploit forks into two processes. The parent uses **ptrace()** to write a new password line, while a child process creates a race condition.
- **Exploitation**: The race condition enables writing a new password line into **/etc/passwd**, bypassing normal permissions.

As a result, a new user with **root privileges** is created in **/etc/passwd**. This user can be accessed using su and the set password, or via SSH. This exploit grants **root** access on the machine.

```
laurie@BornToSecHackMe:~$     gcc -pthread dirty.c -o dirty -lcrypt
laurie@BornToSecHackMe:~$ ./dirty miao
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: miao
Complete line:
admin:fUEQ1lm9fW02:0:0:pwned:/root:/bin/bash

mmap: b7fda000
madvise 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'admin' and the password 'miao'.
laurie@BornToSecHackMe:~$ su admin
Password: miao
admin@BornToSecHackMe:/home/laurie# cd
admin@BornToSecHackMe:~# whoami
admin
admin@BornToSecHackMe:~# id
uid=0(admin) gid=0(root) groups=0(root)
admin@BornToSecHackMe:~# cd /root
admin@BornToSecHackMe:~# ls
README
admin@BornToSecHackMe:~# cat README
CONGRATULATIONS !!!!
To be continued...
```