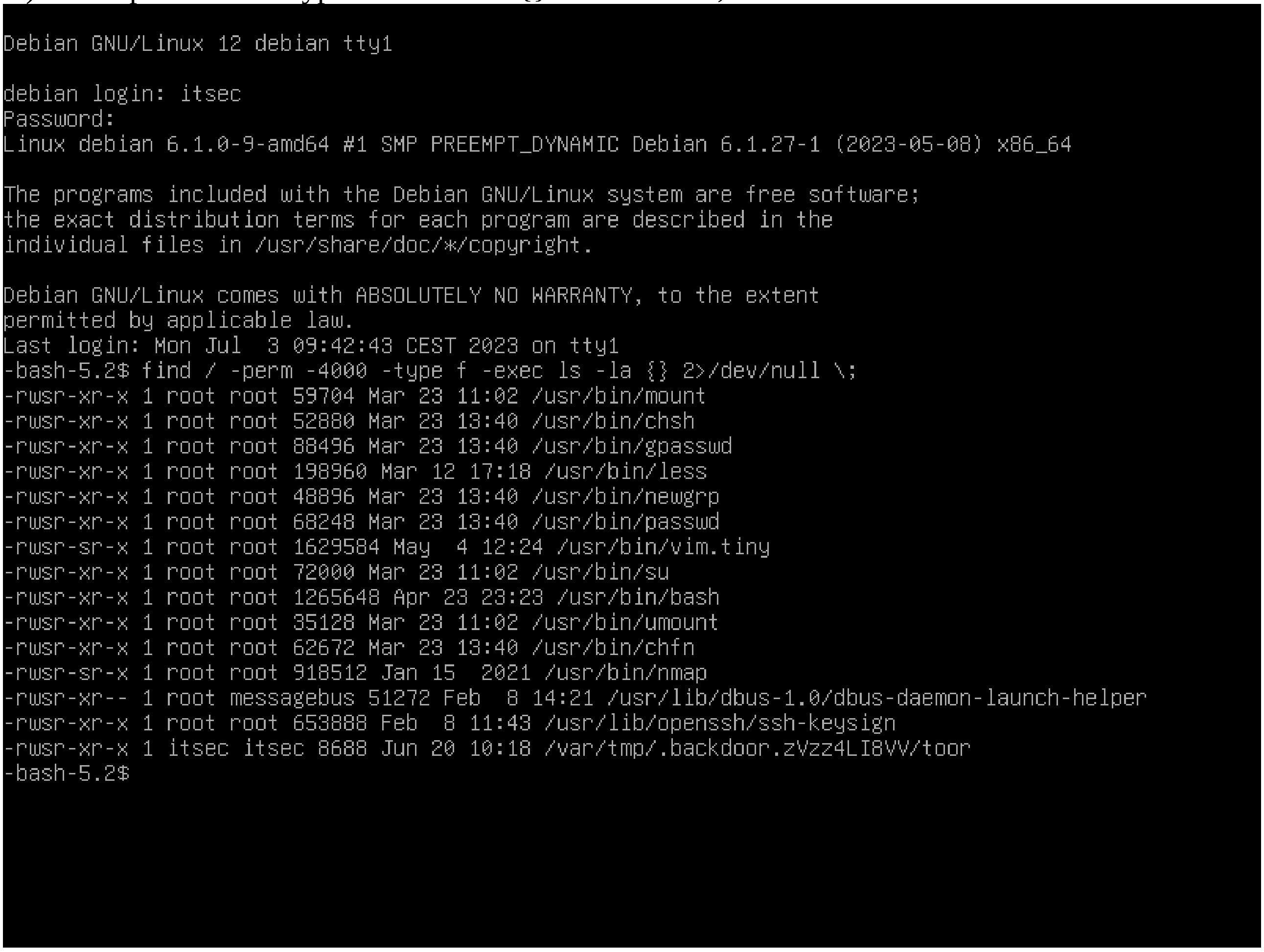
1.
A) find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;



```
Debian GNU/Linux 12 debian tty1

debian login: itsec
Password:
Linux debian 6.1.0-9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1 (2023-05-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jul  3 09:42:43 CEST 2023 on tty1
-bash-5.2$ find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;
-rwsr-xr-x 1 root root 59704 Mar 23 11:02 /usr/bin/mount
-rwsr-xr-x 1 root root 52880 Mar 23 13:40 /usr/bin/chsh
-rwsr-xr-x 1 root root 88496 Mar 23 13:40 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 198960 Mar 12 17:18 /usr/bin/less
-rwsr-xr-x 1 root root 48896 Mar 23 13:40 /usr/bin/newgrp
-rwsr-xr-x 1 root root 68248 Mar 23 13:40 /usr/bin/passwd
-rwsr-sr-x 1 root root 1629584 May  4 12:24 /usr/bin/vim.tiny
-rwsr-xr-x 1 root root 72000 Mar 23 11:02 /usr/bin/su
-rwsr-xr-x 1 root root 1265648 Apr 23 23:23 /usr/bin/bash
-rwsr-xr-x 1 root root 35128 Mar 23 11:02 /usr/bin/umount
-rwsr-xr-x 1 root root 62672 Mar 23 13:40 /usr/bin/chfn
-rwsr-sr-x 1 root root 918512 Jan 15  2021 /usr/bin/nmap
-rwsr-xr-- 1 root messagebus 51272 Feb  8 14:21 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 653888 Feb  8 11:43 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 itsec itsec 8688 Jun 20 10:18 /var/tmp/.backdoor.zVzz4LI8VV/toor
-bash-5.2$
```

B) Vim is primarily used as a text editor. However, if it runs with SUID (Set User ID) privileges, it inherits the permissions of the root user. Consequently, it gains the ability to read all files on the system, including sensitive ones.

For example, executing the command
vim.tiny /etc/shadow
allows Vim to access the „/etc/shadow“ file. Typically, this file is owned by the root user, and its group is set to an administrative group like "shadow." Direct access to the file is restricted to prevent unauthorized users from gathering password hashes of other users.

2.
The typical implementation of the shebang (#!) mechanism introduces a race condition that can lead to security vulnerabilities. The process involves the kernel opening the executable, identifying the #! at the beginning, closing the executable, and then opening the associated interpreter. However, if setuid scripts are allowed, an attacker can exploit this by creating a symbolic link to an existing setuid script, executing it, and manipulating the link before the interpreter opens the script. As a result, most operating systems ignore the setuid bit when a shebang is detected.

One proposed solution is for the kernel to lock the script file until the interpreter opens it. However, due to the avoidance of mandatory locks and the challenges posed by incorporating symbolic links into a proper locking mechanism, this approach is not commonly implemented.

TOCTOU (time-of-check to time-of-use) races are a class of software bugs that stem from race conditions involving the checking and utilization of system states, such as security credentials. These races remain a problem in modern systems, as demonstrated by vulnerabilities discovered in Docker in 2019, allowing unauthorized access to the host platform's filesystem. Exploiting a TOCTOU race condition requires precise timing to ensure that the attacker's actions interleave correctly with the victim's operations.

An alternative solution proposed by the research community is the adoption of transactions within the file system or the operating system kernel. Transactions provide a concurrency control mechanism that can effectively mitigate TOCTOU races.

Many TOCTOU vulnerabilities arise from the lack of synchronization control in the file system API of an operating system. Consequently, finding a comprehensive solution to this problem remains a challenging task.