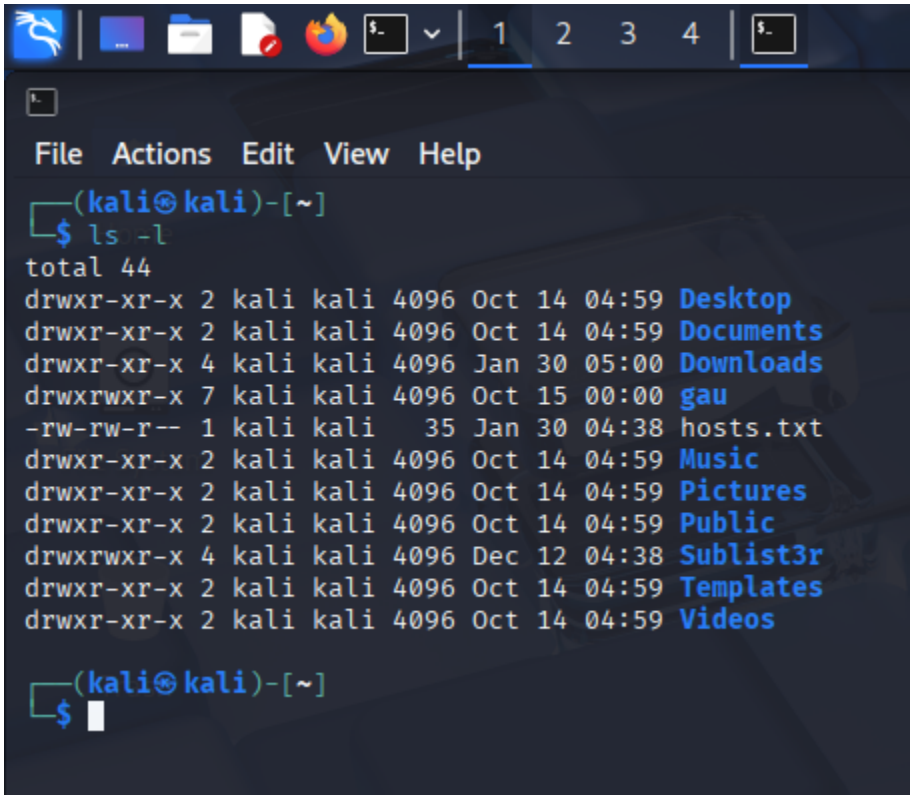
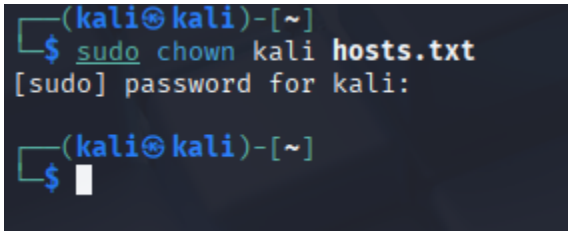
TASK 2

1. View Permissions

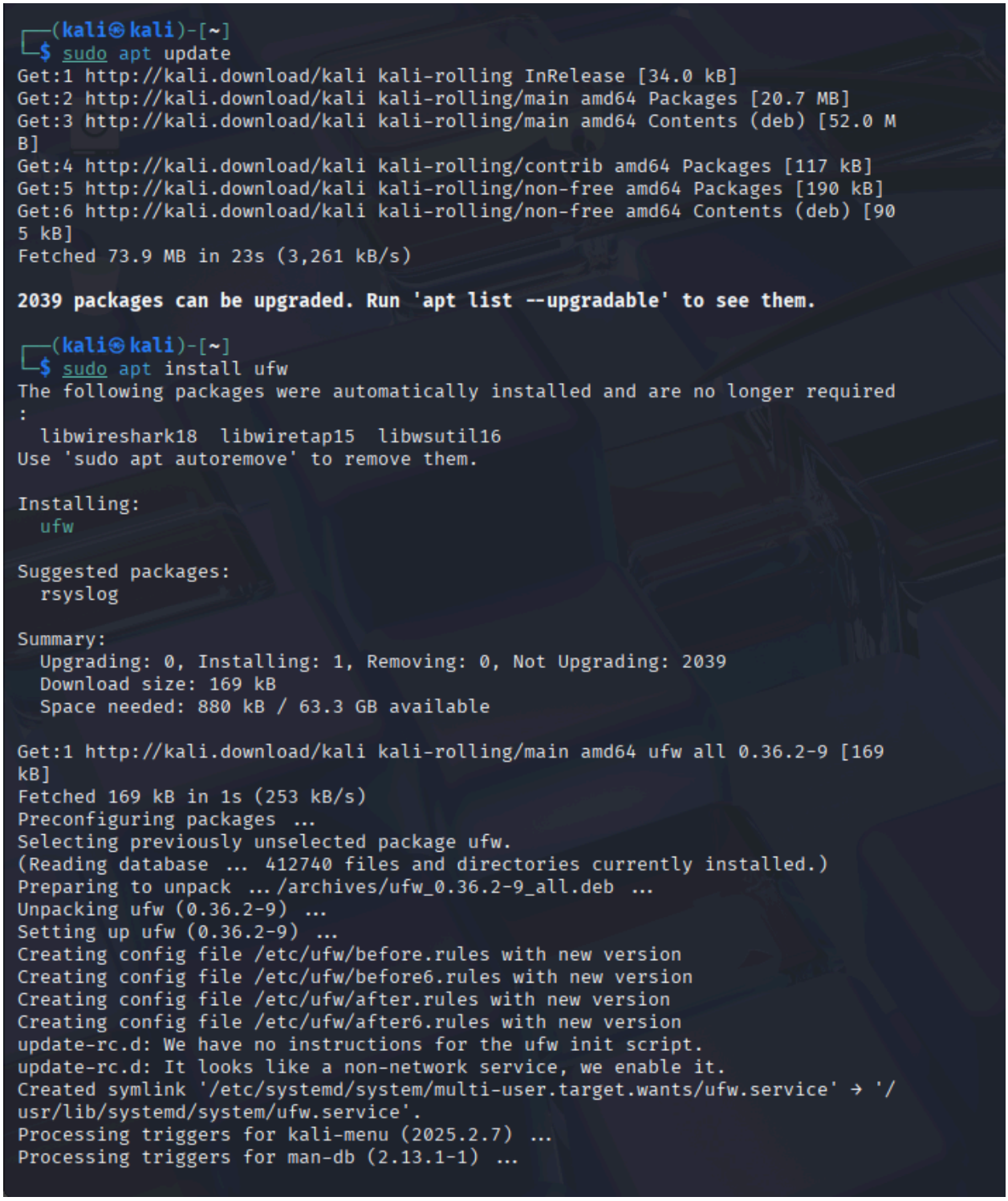


2. Change Permission (chmod):

3. Change ownership (chown)



4. Firewall Configuration

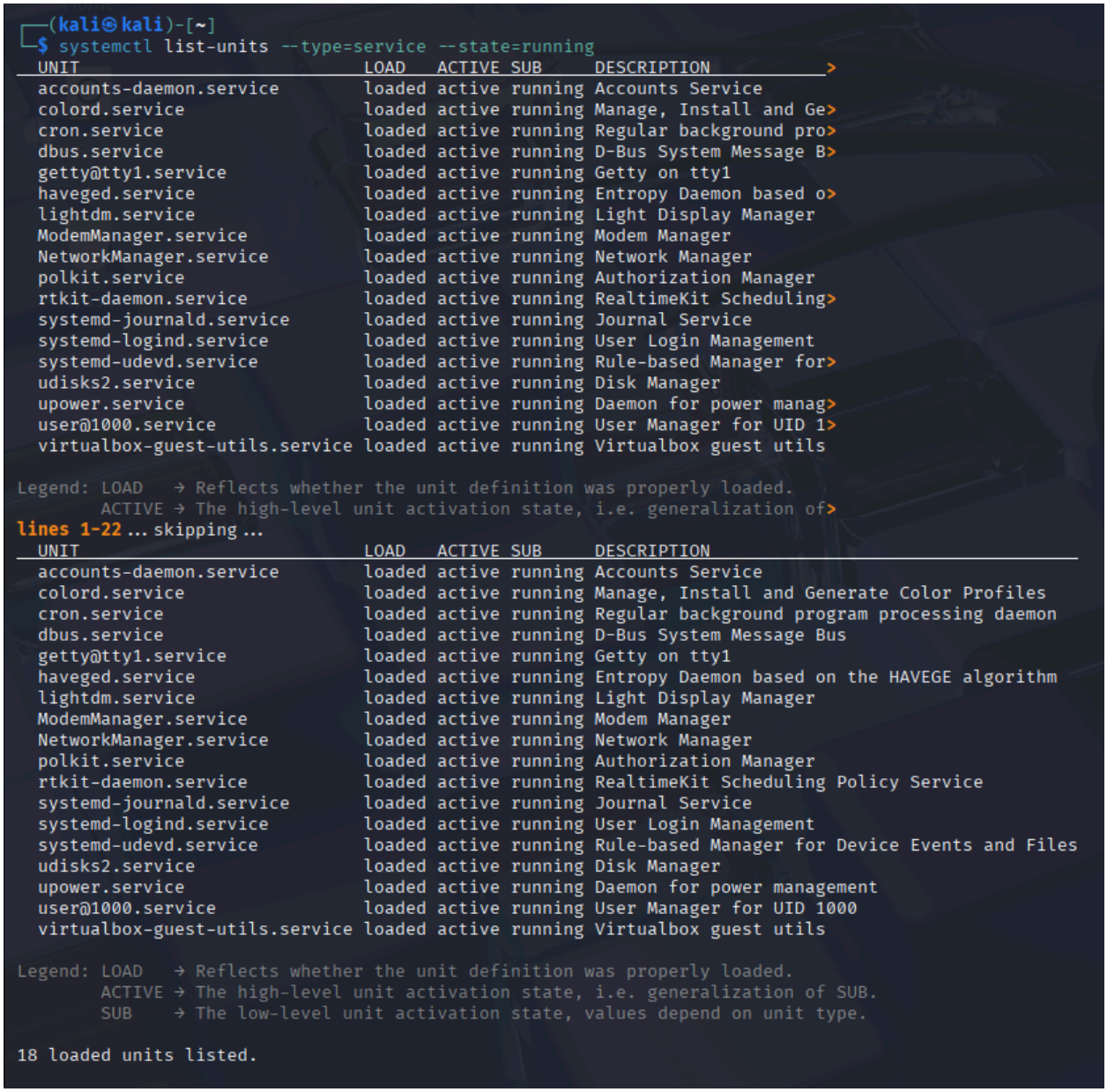


```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status
Status: inactive
```

5. Hardening

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow ssh
Rules updated
Rules updated (v6)

┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

┌──(kali㉿kali)-[~]
└─$
```

6. Identify and disable unnecessary services



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl stop colord.service

┌──(kali㉿kali)-[~]
└─$ sudo systemctl disable colord.service
The unit files have no installation config (WantedBy=, RequiredBy=, UpheldBy=,
Also=, or Alias= settings in the [Install] section, and DefaultInstance= for
template units). This means they are not meant to be enabled or disabled using
 systemctl.

Possible reasons for having these kinds of units are:
• A unit may be statically enabled by being symlinked from another unit's
  .wants/, .requires/, or .upholds/ directory.
• A unit's purpose may be to act as a helper for some other unit which has
  a requirement dependency on it.
• A unit may be started when needed via activation (socket, path, timer,
  D-Bus, udev, scripted systemctl call, ...).
• In case of template units, the unit is meant to be enabled with some
  instance name specified.
```