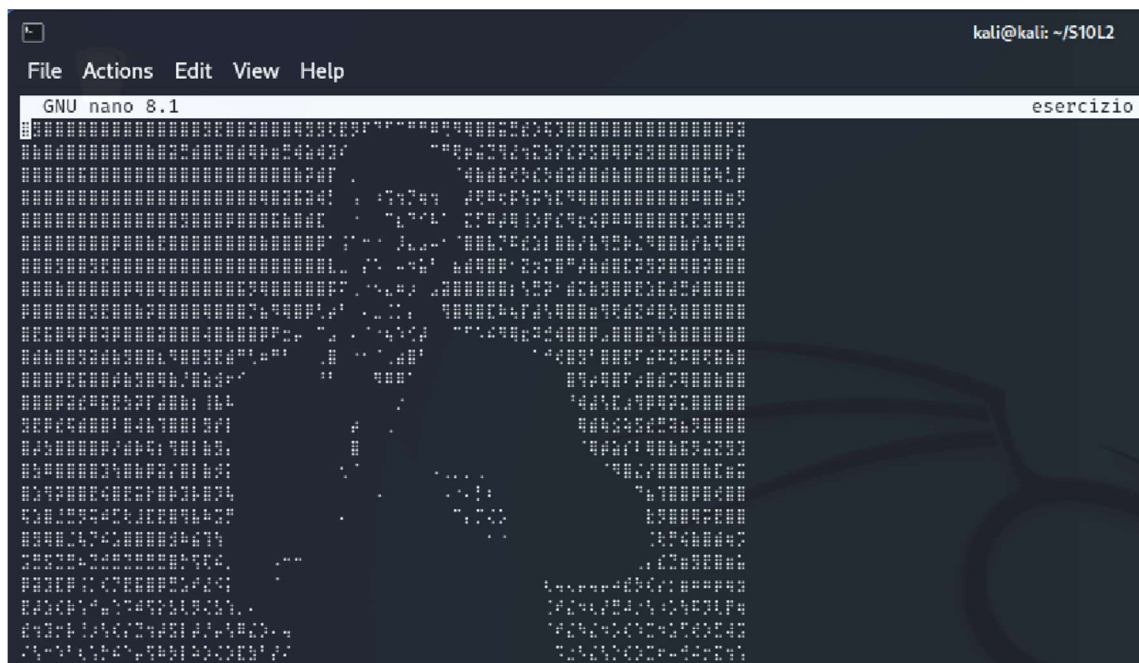


Con il comando “mkdir” andiamo a creare una cartella per l’esercizio.

Con il comando “cd” ci entriamo per creare un file chiamato esercizio usando “nano esercizio”

```
(kali@kali)-[~]  
$ mkdir S10L2  
  
(kali@kali)-[~]  
$ cd S10L2  
  
(kali@kali)-[~/S10L2]  
$ nano esercizio
```



Con ls -l andiamo a visualizzare i permessi attuali del file esercizio

```
(kali@kali)-[~/S10L2]  
$ ls -l  
total 8  
-rw-rw-r-- 1 kali kali 4632 Oct 15 14:25 esercizio
```

Abbiamo permessi di scrittura e lettura sia per l’utente corrente che per il gruppo, mentre gli altri utenti possono solamente leggerlo ma non scriverlo.

Utilizziamo il comando chmod con 640, in modo da rendere modificabile solo per il nostro utente il file:

```
(kali@kali)-[~/S10L2]  
$ chmod 640 esercizio  
  
(kali@kali)-[~/S10L2]  
$ ls -l  
total 8  
-rw-r----- 1 kali kali 4632 Oct 15 14:25 esercizio
```

In questo caso, solamente il nostro utente potrà modificare il file.

Se però noi usassimo chmod 000 nessun utente potrebbe compiere alcuna azione:

```
(kali@kali)-[~/S10L2]
$ chmod 000 esercizio

(kali@kali)-[~/S10L2]
$ ls -l
total 8
----- 1 kali kali 4632 Oct 15 14:25 esercizio

(kali@kali)-[~/S10L2]
$ nano esercizio

(kali@kali)-[~/S10L2]
$ cat esercizio
cat: esercizio: Permission denied
```

**[ Error reading esercizio: Permission denied ]**

^T	Execute	^C	Location	M-U	Undo	M-A
^J	Justify	^/_	Go To Line	M-F	Redo	M-6

Proviamo anche a sovrascrivere il contenuto:

```
(kali@kali)-[~/S10L2]
$ echo "You know the rules, and so do I (Do I)" > esercizio
zsh: permission denied: esercizio
```

Situazione simulata:

abbiamo una cartella chiamata Test, contenente due file, un eseguibile e un file README di istruzioni per l'eseguibile stesso.

```
(kali@kali)-[~/S10L2]
$ mkdir Test

(kali@kali)-[~/S10L2]
$ cd Test

(kali@kali)-[~/S10L2/Test]
$ nano README

(kali@kali)-[~/S10L2/Test]
$ nano eseguibile

(kali@kali)-[~/S10L2/Test]
$ tree
.
├── eseguibile
└── README

1 directory, 2 files
```

Impostiamo che la cartella abbia tutti i permessi per tutti gli utenti:

```
(kali@kali)-[~/S10L2]
$ ls -l
total 12
----- 1 kali kali 4632 Oct 15 14:25 esercizio
drwxrwxrwx 2 kali kali 4096 Oct 15 15:12 Test
```

Andiamo ad applicare a README i permessi solamente di lettura da parte del utente corrente e del gruppo, mentre tutti gli altri utenti non possono neanche visualizzare il contenuto, con chmod 440.

Applichiamo i permessi di esecuzione solo ad utente corrente e gruppo al file eseguibile con chmod 110:

```
(kali@kali)-[~/S10L2/Test]
$ chmod 440 README

(kali@kali)-[~/S10L2/Test]
$ chmod 110 eseguibile

(kali@kali)-[~/S10L2/Test]
$ ls -l
total 12
--x--x-- 1 kali kali 58 Oct 15 15:00 eseguibile
-r--r-- 1 kali kali 4170 Oct 15 14:58 README
```

Se provassimo a leggere eseguibile questa sarebbe la risposta:

```
[ Error reading eseguibile: Permission denied ]
Execute ^C Location M-U Undo M-A
```

Mentre se usassimo il comando sudo nano riusciremmo comunque a visualizzare:

```
(kali@kali)-[~/S10L2/Test]
$ sudo nano eseguibile
[sudo] password for kali:
```

```
kali@kali: ~/S10L2/Test
File Actions Edit View Help
GNU nano 8.1 eseguibile
Thank you Mario! But our executable is in another castle!
|
```

Il file readme invece, rimane visualizzabile ma non editabile, se non con permessi di root:

```
(kali@kali)-[~/S10L2/Test]
$ echo "prova di modifica" > README
zsh: permission denied: README
```

Ma cosa succederebbe se ad esempio il file README garantisse tutti i permessi a chiunque ma la cartella non permettesse ad esempio, la modifica?

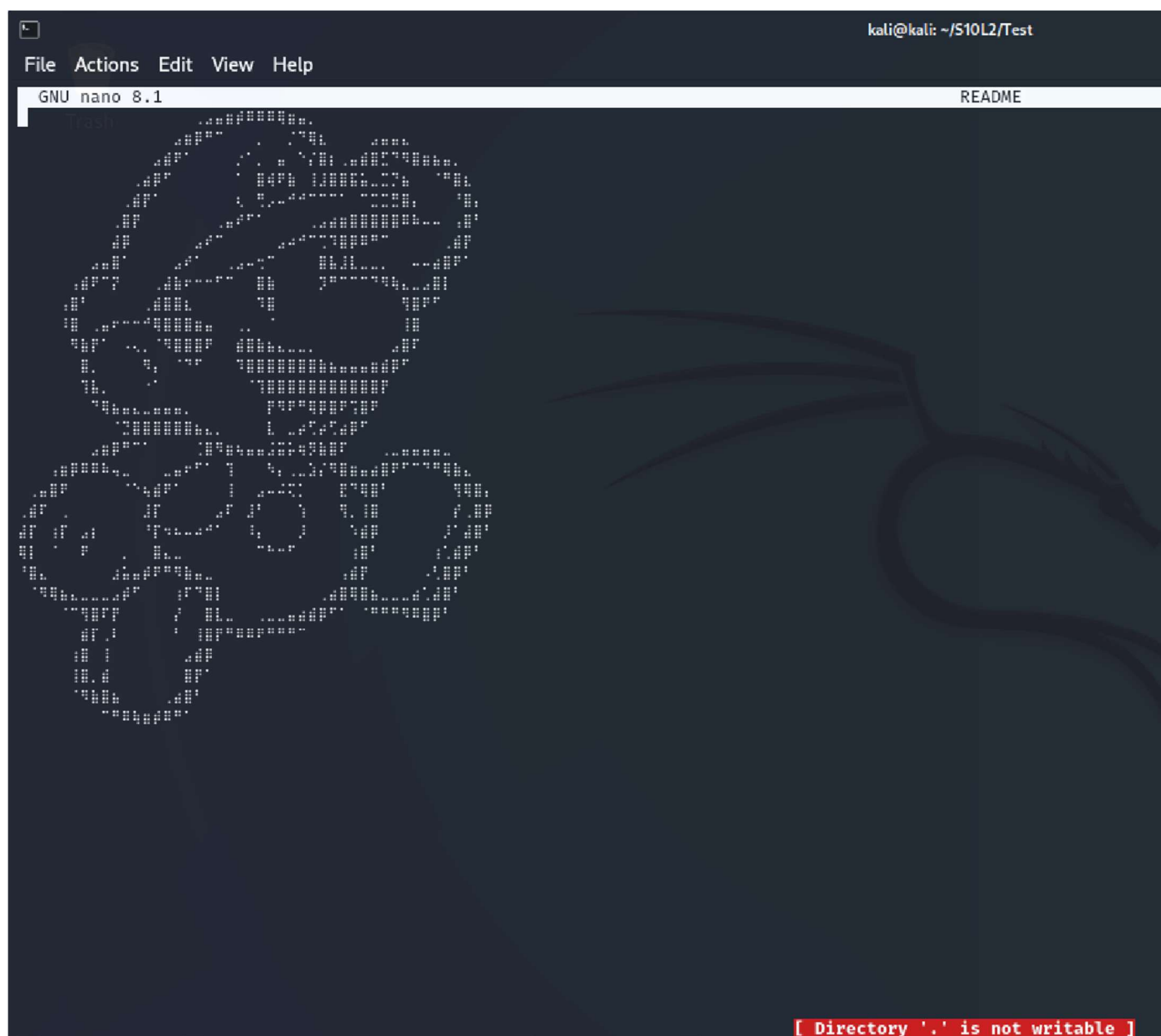
Impostiamo il file README con chmod 777 e la cartella Test con chmod 555:

```
(kali@kali)-[~/S10L2]
$ chmod 555 Test

(kali@kali)-[~/S10L2]
$ cd Test

(kali@kali)-[~/S10L2/Test]
$ chmod 777 README

(kali@kali)-[~/S10L2/Test]
$ nano README
```



Questo perché, nonostante il file garantisca i permessi, sono i permessi della cartella a influenzare i file contenuti