

# TASK 4

In this digital word security and privacy is important because data is the nowadays assert. To protect our data from the third party members and attackers, we need to change the plain text into a cipher text. Changing the plain text into cipher text can achieve with several process like encryption and hashing.

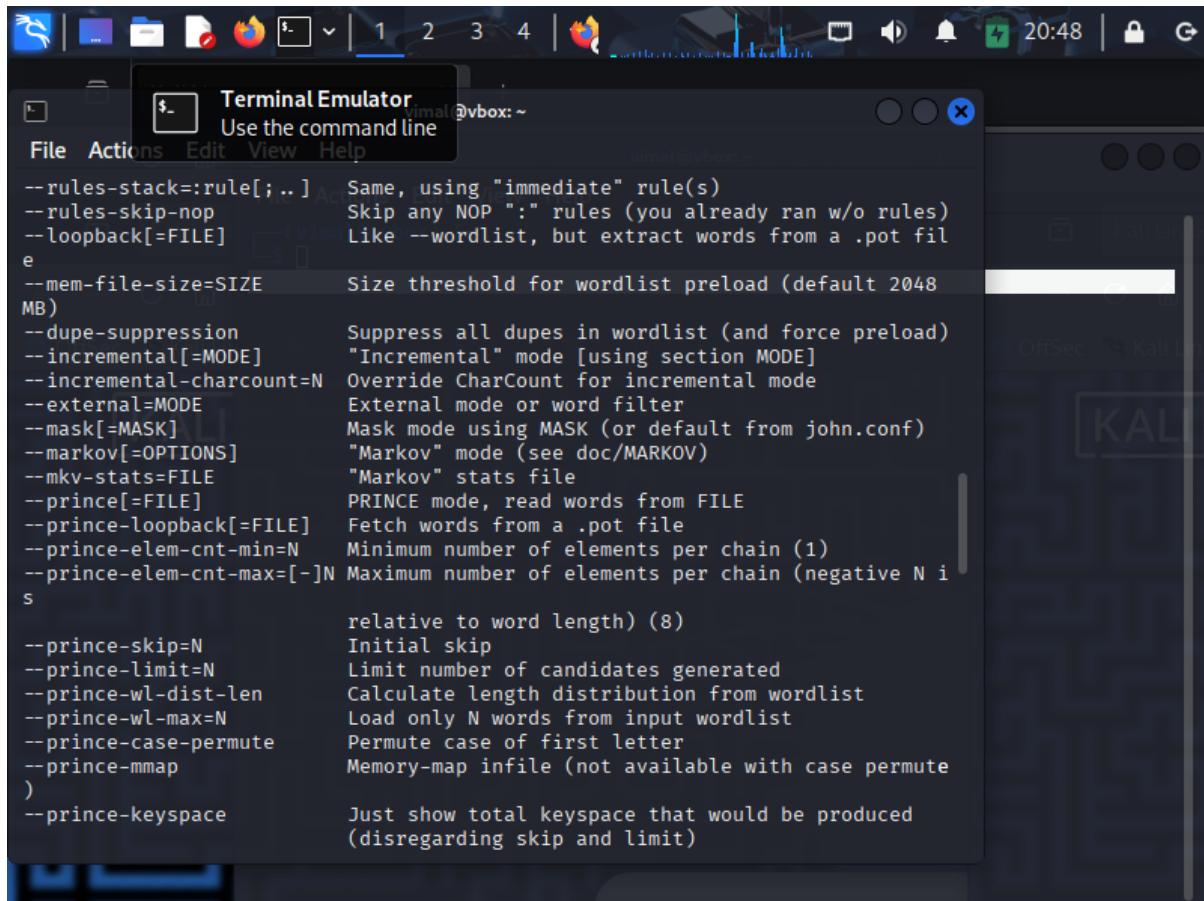
## HASHING

Hashing is process of changing the text into a sequence of number majorly in hexadecimal values, this process is fixed we cant able to reverse it, this thing is useful to store the credentials at server. Even the server doesn't know the original text or script. Example **fad89cd3d010b686f0fb8aaf22d79576d589d29f648b42d917b7f5cafe6c0ed3** Is the hash value of **WannaCry ransomware SHA256 hash.txt**

## ENCRYPTION

Encryption is also do the same things like hashing, changing the plain text into a cipher text. But the working mechanism is makes the difference here, in hashing there is no reverse process but in encryption the both side reverse process may occur. The changing process is achieved by the crypto keys like **public key and private key**.

- **HASHING** -ONE WAY PROCESS
- **ENCRYPTION** -TWO WAY PROCESS(crypto keys)



```
--rules-stack=:rule[; ..] Same, using "immediate" rule(s)
--rules-skip-nop Skip any NOP ":" rules (you already ran w/o rules)
--loopback[=FILE] Like --wordlist, but extract words from a .pot fil
e
--mem-file-size=SIZE Size threshold for wordlist preload (default 2048
MB)
--dupe-suppression Suppress all dupes in wordlist (and force preload)
--incremental[=MODE] "Incremental" mode [using section MODE]
--incremental-charcount=N Override CharCount for incremental mode
--external=MODE External mode or word filter
--mask[=MASK] Mask mode using MASK (or default from john.conf)
--markov[=OPTIONS] "Markov" mode (see doc/MARKOV)
--mkv-stats=FILE "Markov" stats file
--prince[=FILE] PRINCE mode, read words from FILE
--prince-loopback[=FILE] Fetch words from a .pot file
--prince-elem-cnt-min=N Minimum number of elements per chain (1)
--prince-elem-cnt-max=[-]N Maximum number of elements per chain (negative N i
s
relative to word length) (8)
--prince-skip=N Initial skip
--prince-limit=N Limit number of candidates generated
--prince-wl-dist-len Calculate length distribution from wordlist
--prince-wl-max=N Load only N words from input wordlist
--prince-case-permute Permute case of first letter
--prince-mmap Memory-map infile (not available with case permute
)
--prince-keyspace Just show total keyspace that would be produced
(disregarding skip and limit)
```

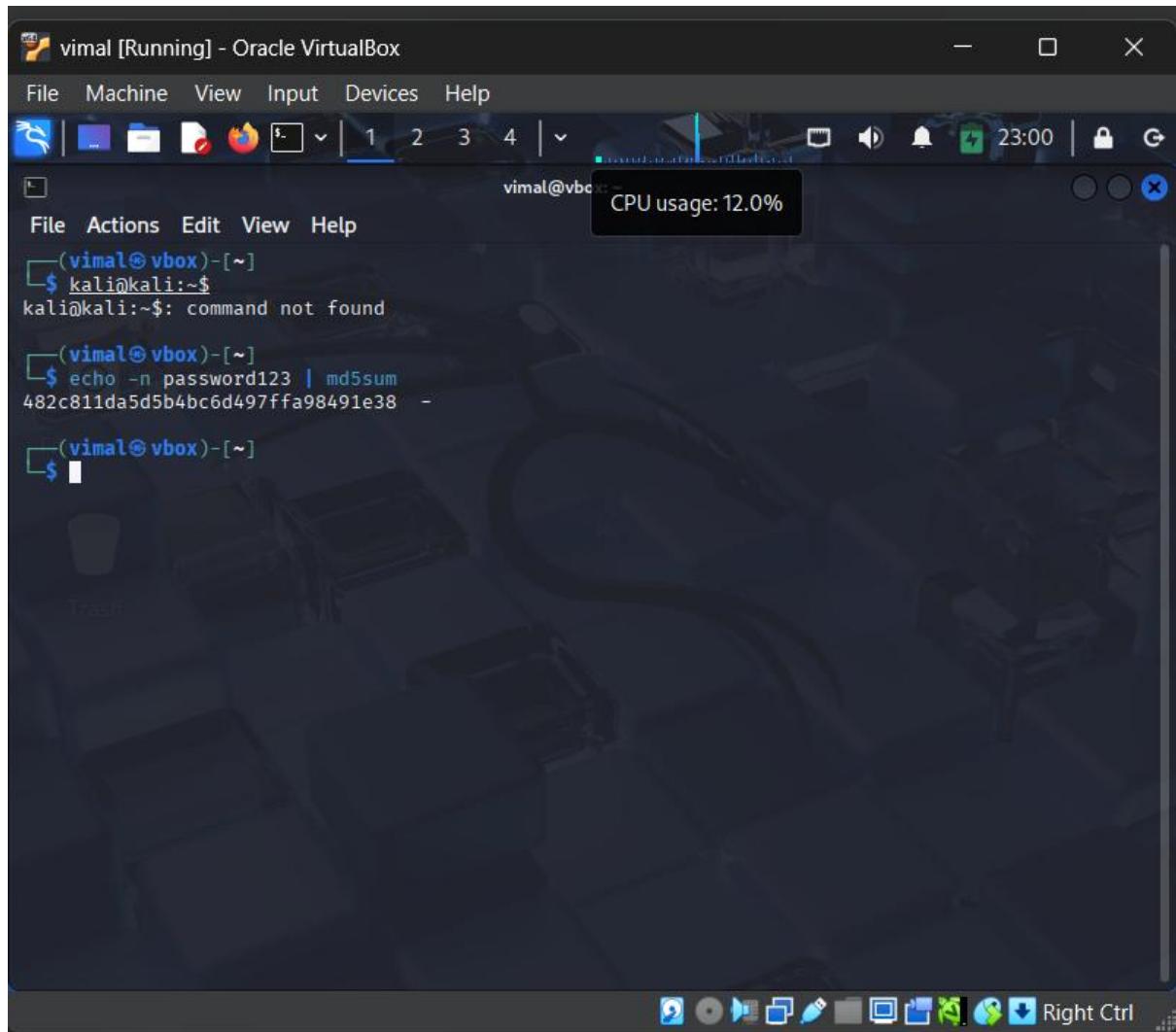
I have installed the john ripper in my vm, kali linux runn it in mu vm terminal and run the following commands

- sudo apt update : update the firmware.
- sudo apt install john -y : install the John ripper
- john --help : confirm installation
- echo -n password123 | md5sum :Create a harsh using text

Here the the password123 is the word to hashing

- **echo** : to print
- **-n** : to decleare there is no new line
- **Password123**: this is the text
- **Md5sum** : it is the command to hash the value

482c811da5d5b4bc6d497ffa98491e38 -this is the hash value



The screenshot shows a terminal window titled "vimal [Running] - Oracle VirtualBox". The terminal is running on a Kali Linux system. The user has entered the following commands:

```
vimal@vbc:~$ sudo apt update
vimal@vbc:~$ sudo apt install john -y
vimal@vbc:~$ john --help
vimal@vbc:~$ echo -n password123 | md5sum
482c811da5d5b4bc6d497ffa98491e38
vimal@vbc:~$
```

A tooltip "CPU usage: 12.0%" is visible above the terminal window. The desktop environment includes icons for file, machine, view, input, devices, help, and a browser. The taskbar at the bottom shows various application icons.