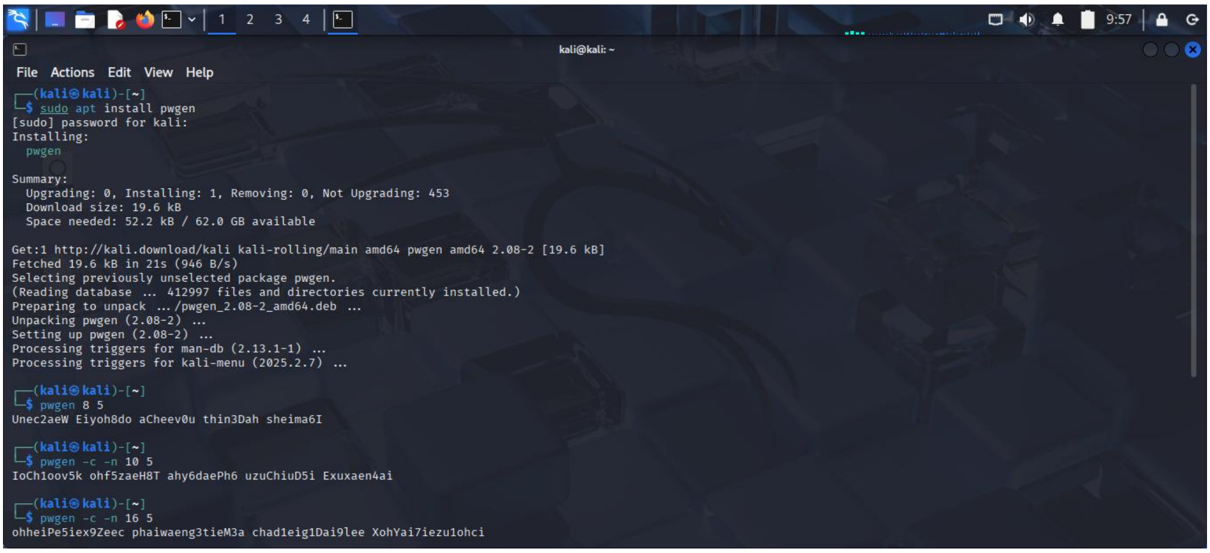
# Task 6

## Task : Create a Strong Password and Evaluate Its Strength.

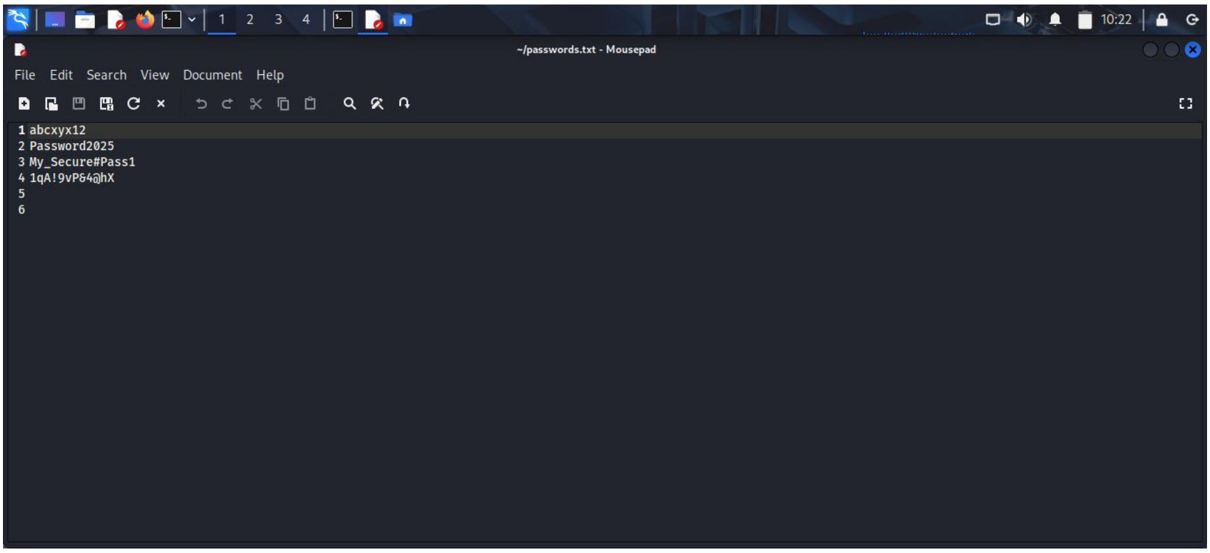
*Step 1:* Create multiple passwords with varying complexity

Generate examples :



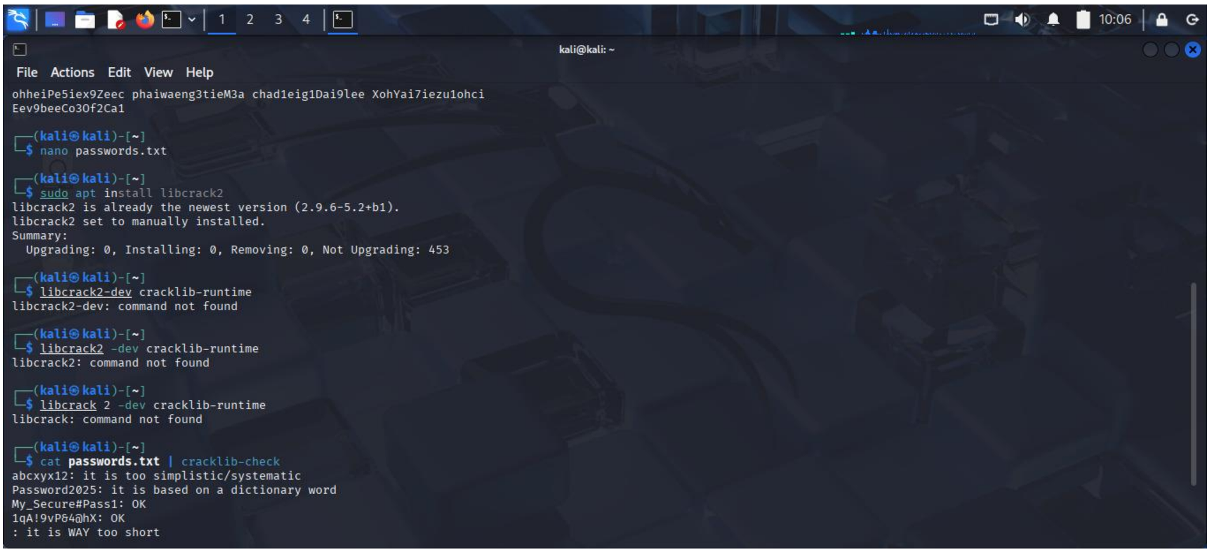
```
┌──(kali㉿kali)-[~]
└─$ sudo apt install pwgen
[sudo] password for kali:
Installing:
  pwgen

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 453
  Download size: 19.6 kB
  Space needed: 52.2 kB / 62.0 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 pwgen amd64 2.08-2 [19.6 kB]
Fetched 19.6 kB in 21s (946 B/s)
Selecting previously unselected package pwgen.
(Reading database ... 412997 files and directories currently installed.)
Preparing to unpack .../pwgen_2.08-2_amd64.deb ...
Unpacking pwgen (2.08-2) ...
Setting up pwgen (2.08-2) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.2.7) ...

┌──(kali㉿kali)-[~]
└─$ pwgen 8 5
Unec2aeW Eiyoh8do aCheev0u thin3Dah sheima6I

┌──(kali㉿kali)-[~]
└─$ pwgen -c -n 10 5
IoCh1oov5k ohf5zaeH8T ahy6daePh6 uzuChiuD5i Exuxaen4ai

┌──(kali㉿kali)-[~]
└─$ pwgen -c -n 16 5
ohheiPe5iex9Zeec phaiwaeng3tieM3a chad1eig1Dai9lee XohYai7iezu1ohci
```

*Step 2:* Use uppercase, lowercase, numbers, symbols, and very length


1 2 3 4
10:22
~/passwords.txt - Mousepad
File Edit Search View Document Help

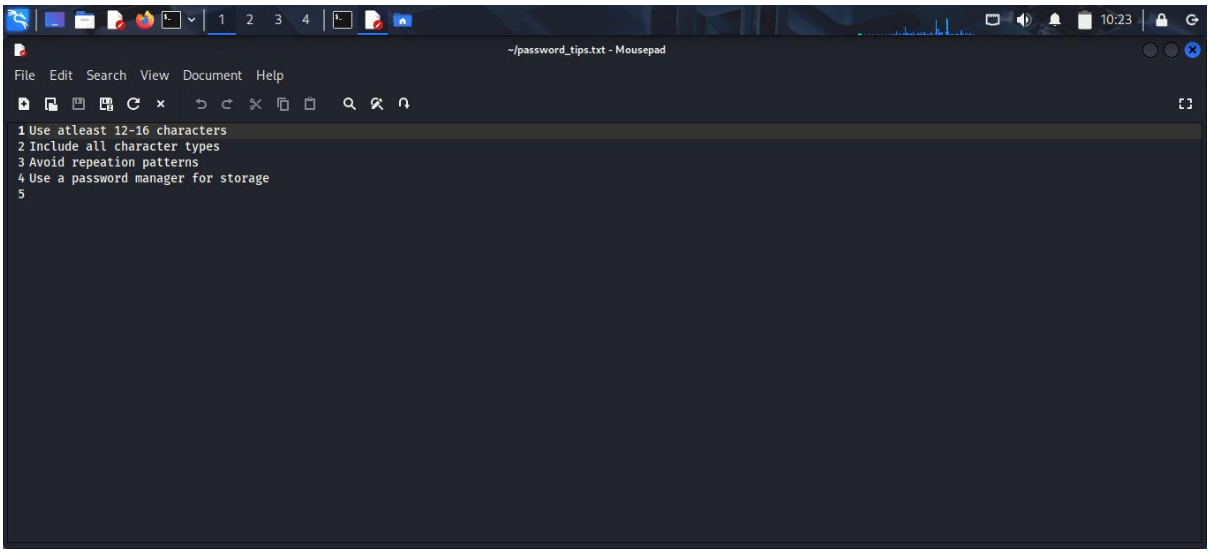
```
abcxyx12
Password2025
My_Secure#Pass1
1qA!9vP&4@hX
```

## *Step 3:* Test each password  on a password strength checker



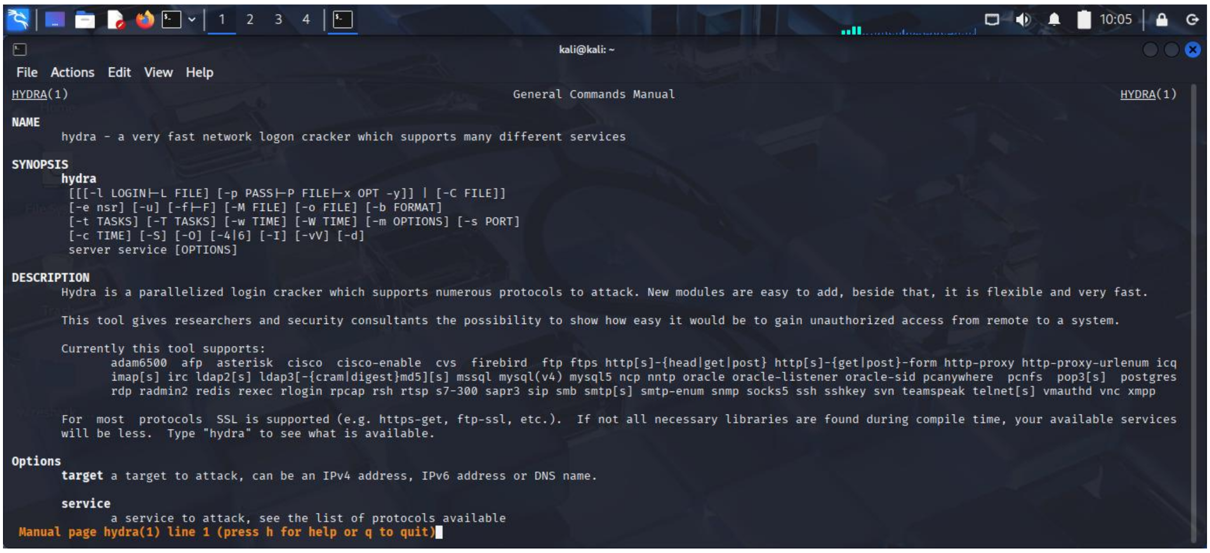
```
ohheiPe5iex9Zeec phaiwaeng3tieM3a chad1eig1Dai9lee XohYai7iezu1ohci
Eev9beeCo3Of2Ca1

┌──(kali㉿kali)-[~]
└─$ nano passwords.txt

┌──(kali㉿kali)-[~]
└─$ sudo apt install libcrack2
libcrack2 is already the newest version (2.9.6-5.2+b1).
libcrack2 set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 453

┌──(kali㉿kali)-[~]
└─$ libcrack2-dev cracklib-runtime
libcrack2-dev: command not found

┌──(kali㉿kali)-[~]
└─$ libcrack2 -dev cracklib-runtime
libcrack2: command not found

┌──(kali㉿kali)-[~]
└─$ libcrack 2 -dev cracklib-runtime
libcrack: command not found

┌──(kali㉿kali)-[~]
└─$ cat passwords.txt | cracklib-check
abcxyx12: it is too simplistic/systematic
Password2025: it is based on a dictionary word
My_Secure#Pass1: OK
1qA!9vP&4@hX: OK
: it is WAY too short
```

## *Step 4:* Write down tips learned

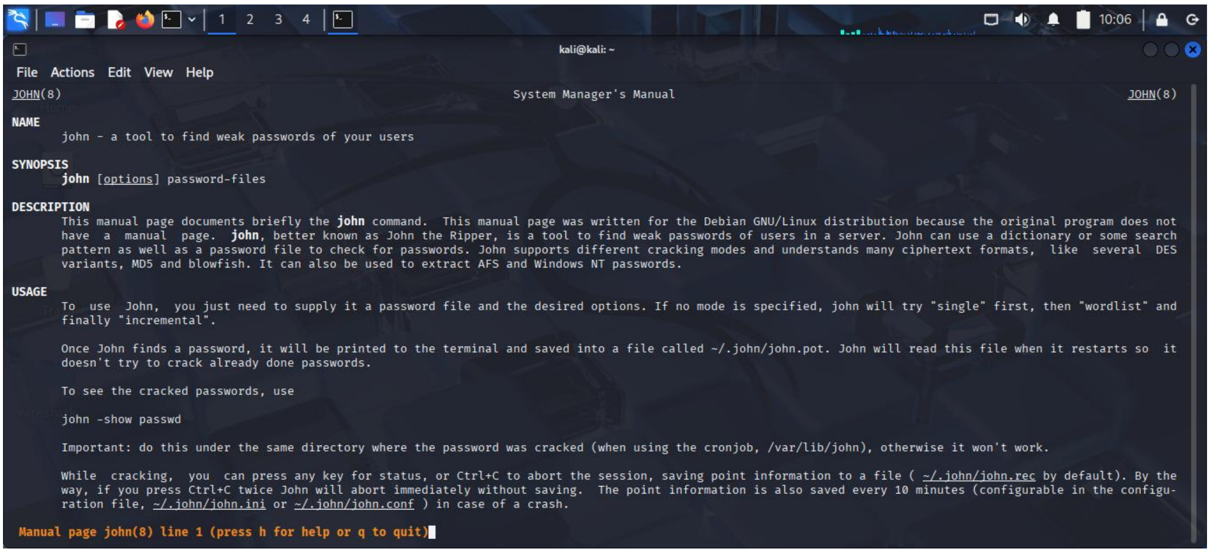

~/password_tips.txt - Mousepad
File Edit Search View Document Help
10:23

```
Use atleast 12-16 characters
Include all character types
Avoid repeation patterns
Use a password manager for storage

```

## *Step 5:* Research common password attack

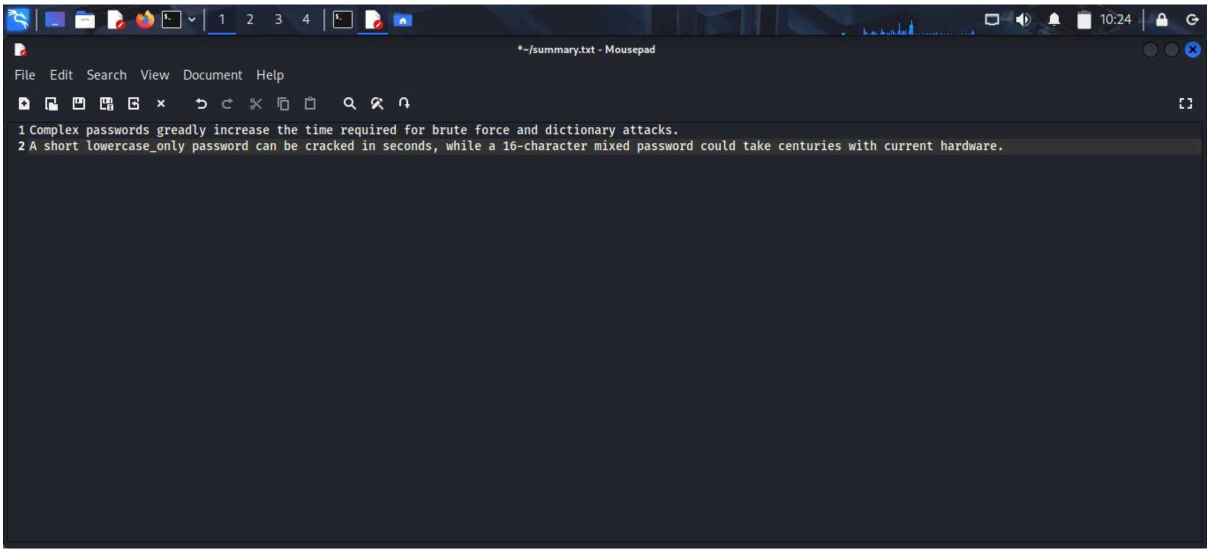


```
HYDRA(1)                                  General Commands Manual                                  HYDRA(1)

NAME
       hydra - a very fast network logon cracker which supports many different services

SYNOPSIS
       hydra
        [[[-l LOGIN|-L FILE] [-p PASS|-P FILE|-x OPT -y]] | [-C FILE]]
        [-e nsr] [-u] [-f|-F] [-M FILE] [-o FILE] [-b FORMAT]
        [-t TASKS] [-T TASKS] [-w TIME] [-W TIME] [-m OPTIONS] [-s PORT]
        [-c TIME] [-S] [-O] [-4|6] [-I] [-vV] [-d]
        server service [OPTIONS]

DESCRIPTION
       Hydra is a parallelized login cracker which supports numerous protocols to attack. New modules are easy to add, beside that, it is flexible and very fast.

       This tool gives researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.

       Currently this tool supports:
              adam6500  afp  asterisk  cisco  cisco-enable  cvs  firebird  ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq
              imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql(v4) mysql5 ncp nntp oracle oracle-listener oracle-sid pcanywhere  pcnfs  pop3[s]  postgres
              rdp radmin2 redis rexec rlogin rpcap rsh rtsp s7-300 sapr3 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

       For  most  protocols  SSL is supported (e.g. https-get, ftp-ssl, etc.).  If not all necessary libraries are found during compile time, your available services
       will be less.  Type "hydra" to see what is available.

Options
       target a target to attack, can be an IPv4 address, IPv6 address or DNS name.

       service
              a service to attack, see the list of protocols available
 Manual page hydra(1) line 1 (press h for help or q to quit)
```



```
JOHN(8)                                   System Manager's Manual                                   JOHN(8)

NAME
       john - a tool to find weak passwords of your users

SYNOPSIS
       john [options] password-files

DESCRIPTION
       This manual page documents briefly the john command.  This manual page was written for the Debian GNU/Linux distribution because the original program does not
       have  a  manual  page.  john, better known as John the Ripper, is a tool to find weak passwords of users in a server. John can use a dictionary or some search
       pattern as well as a password file to check for passwords. John supports different cracking modes and understands many ciphertext formats,  like  several  DES
       variants, MD5 and blowfish. It can also be used to extract AFS and Windows NT passwords.

USAGE
       To  use  John,  you just need to supply it a password file and the desired options. If no mode is specified, john will try "single" first, then "wordlist" and
       finally "incremental".

       Once John finds a password, it will be printed to the terminal and saved into a file called ~/.john/john.pot. John will read this file when it restarts so  it
       doesn't try to crack already done passwords.

       To see the cracked passwords, use

       john -show passwd

       Important: do this under the same directory where the password was cracked (when using the cronjob, /var/lib/john), otherwise it won't work.

       While  cracking,  you  can press any key for status, or Ctrl+C to abort the session, saving point information to a file ( ~/.john/john.rec by default). By the
       way, if you press Ctrl+C twice John will abort immediately without saving.  The point information is also saved every 10 minutes (configurable in the configu-
       ration file, ~/.john/john.ini or ~/.john/john.conf ) in case of a crash.

 Manual page john(8) line 1 (press h for help or q to quit)
```

## *Step 6:* Summarize how password complexity affect security



```
Complex passwords greadly increase the time required for brute force and dictionary attacks.
A short lowercase_only password can be cracked in seconds, while a 16-character mixed password could take centuries with current hardware.
```