

Task 6 : Create a Strong Password and Evaluate Its Strength.

Anisha Khairnar (anishakhairnar284@gmail.com)

Cybersecurity Intern.

Introduction :

In this task, we create passwords with different complexity levels, test their strength using online tools, analyze the results, and summarize best practices for building secure passwords.

1] Installed cracklib-runtime

```
(kali㉿kali)-[~/Task6_Passwords]
$ sudo apt install pwgen cracklib-runtime -y
cracklib-runtime is already the newest version (2.9.6-5.2+b1).
cracklib-runtime set to manually installed.
The following packages were automatically installed and are no longer required:
  dialign      emboss-lib      primer3      python3-pyinstaller-hooks-contrib
  emboss-data  libhpdf-2.3.0  python3-packaging-whl  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Installing:
  pwgen

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1084
  Download size: 19.6 kB
  Space needed: 52.2 kB / 55.4 GB available

Get:1 http://mirrors.esto.network/kali kali-rolling/main amd64 pwgen amd64 2.08-2 [19.6 kB]
Fetched 19.6 kB in 1s (17.0 kB/s)
Selecting previously unselected package pwgen.
(Reading database ... 431245 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 (2025.3.0) ...
```

2] Generate a few random passwords and stored in passwords.txt

```
(kali㉿kali)-[~]
$ mkdir -p ~/Task6_Passwords && cd ~/Task6_Passwords

(kali㉿kali)-[~/Task6_Passwords]
$ cat > passwords.txt << 'EOF'
heredoc> hello
heredoc> Hello123
heredoc> H3ll0@2025!
heredoc> MyS3cur##Passw0rd 2025!!
heredoc> EOF
```

3] Local strength check with cracklib-check

```
(kali@kali)-[~/Task6_Passwords]
$ pwgen -sy 16 5 > generated.txt

(kali@kali)-[~/Task6_Passwords]
$ openssl rand -base64 18 >> generated.txt

(kali@kali)-[~/Task6_Passwords]
$ # For each password in passwords.txt, show cracklib-check output
while IFS= read -r p; do
    echo -n "$p: "
    echo "$p" | cracklib-check
done < passwords.txt

hello: hello: it is too short
Hello123: Hello123: it is based on a dictionary word
H3llo@2025!: H3llo@2025!: OK
MyS3cur##Passw0rd_2025 !! : MyS3cur##Passw0rd_2025 !! : OK
```

4] Compute approximate entropy (bits) for each password

```
(kali@kali)-[~/Task6_Passwords]
$ python3 - <<'PY'
import math
charset = 94 # typical printable ASCII set (26+26+10+32)
with open('passwords.txt') as f:
    for pw in f:
        pw = pw.strip()
        if not pw: continue
        entropy = len(pw) * math.log2(charset)
        print(f"{pw:30} len={len(pw):2} entropy*={entropy:.2f} bits")
PY
hello                               len= 5  entropy*32.77 bits
Hello123                           len= 8  entropy*52.44 bits
H3llo@2025!                        len=11  entropy*72.10 bits
MyS3cur##Passw0rd_2025 !!          len=24  entropy*157.31 bits
```

Higher bits = stronger (≥ 80 bits is generally strong for passwords/passphrases).

Online checks for demonstration : Checked on passwordmeter.com

1] Very Weak password: hello

The Password Meter

Test Your Password		Minimum Requirements				
Password:	<div>•••••</div>	<ul style="list-style-type: none">Minimum 8 characters in lengthContains 3/4 of the following items:<ul style="list-style-type: none">Uppercase LettersLowercase LettersNumbersSymbols				
Hide:	<input checked="" type="checkbox"/>					
Score:	<div>4%</div>					
Complexity:	Very Weak					

Additions		Type	Rate	Count	Bonus
✖	Number of Characters	Flat	$+(n^4)$	<div>5</div>	+ 20
✖	Uppercase Letters	Cond/Incr	$+\frac{(len-n)^2}{2}$	<div>0</div>	0
☑	Lowercase Letters	Cond/Incr	$+\frac{(len-n)^2}{2}$	<div>5</div>	0
✖	Numbers	Cond	$+(n^4)$	<div>0</div>	0
✖	Symbols	Flat	$+(n^6)$	<div>0</div>	0
✖	Middle Numbers or Symbols	Flat	$+(n^2)$	<div>0</div>	0
✖	Requirements	Flat	$+(n^2)$	<div>1</div>	0
Deductions					
⚠	Letters Only	Flat	$-n$	<div>5</div>	- 5
☑	Numbers Only	Flat	$-n$	<div>0</div>	0
⚠	Repeat Characters (Case Insensitive)	Comp	-	<div>2</div>	- 3
☑	Consecutive Uppercase Letters	Flat	$-(n^2)$	<div>0</div>	0
⚠	Consecutive Lowercase Letters	Flat	$-(n^2)$	<div>4</div>	- 8

2] Weak- Medium password: Hello123

The Password Meter

Test Your Password		Minimum Requirements			
Password:	<div>••••••••</div>	<ul style="list-style-type: none">Minimum 8 characters in lengthContains 3/4 of the following items:<ul style="list-style-type: none">Uppercase LettersLowercase LettersNumbersSymbols			
Hide:	<input checked="" type="checkbox"/>				
Score:	<div>37%</div>				
Complexity:	Weak				

Additions		Type	Rate	Count	Bonus
✓	Number of Characters	Flat	$+(n^4)$	8	+ 32
✗	Uppercase Letters	Cond/ Incr	$+\frac{((len-n)^2)}{2}$	0	0
✓	Lowercase Letters	Cond/ Incr	$+\frac{((len-n)^2)}{2}$	5	+ 6
✓	Numbers	Cond	$+(n^4)$	3	+ 12
✗	Symbols	Flat	$+(n^6)$	0	0
✓	Middle Numbers or Symbols	Flat	$+(n^2)$	2	+ 4
✗	Requirements	Flat	$+(n^2)$	3	0
Deductions					
✓	Letters Only	Flat	$-n$	0	0
✓	Numbers Only	Flat	$-n$	0	0
⚠	Repeat Characters (Case Insensitive)	Comp	-	2	- 2
✓	Consecutive Uppercase Letters	Flat	$-(n^2)$	0	0
⚠	Consecutive Lowercase Letters	Flat	$-(n^2)$	4	- 8

3]Medium – Strong : H3llo@2025!

Test Your Password		Minimum Requirements	
Password:	••••••••	<ul style="list-style-type: none"> Minimum 8 characters in length Contains 3/4 of the following items: <ul style="list-style-type: none"> Uppercase Letters Lowercase Letters Numbers Symbols 	
Hide:	<input checked="" type="checkbox"/>		
Score:	100%		
Complexity:	Very Strong		

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	$+(n^4)$	11	+ 44
Uppercase Letters	Cond/ Incr	$+(len-n)^2$	1	+ 20
Lowercase Letters	Cond/ Incr	$+(len-n)^2$	3	+ 16
Numbers	Cond	$+(n^4)$	5	+ 20
Symbols	Flat	$+(n^6)$	2	+ 12
Middle Numbers or Symbols	Flat	$+(n^2)$	6	+ 12
Requirements	Flat	$+(n^2)$	5	+ 10

Deductions	Type	Rate	Count	Bonus
Letters Only	Flat	-n	0	0
Numbers Only	Flat	-n	0	0
Repeat Characters (Case Insensitive)	Comp	-	4	- 1
Consecutive Uppercase Letters	Flat	$-(n^2)$	0	0
Consecutive Lowercase Letters	Flat	$-(n^2)$	2	- 4
Consecutive Numbers	Flat	$-(n^2)$	3	- 6
Sequential Letters (3+)	Flat	$-(n^3)$	0	0
Sequential Numbers (3+)	Flat	$-(n^3)$	0	0
Sequential Symbols (3+)	Flat	$-(n^3)$	0	0

4] Very Strong : MyS3cur3#Passw0rd_2025!!

Test Your Password		Minimum Requirements	
Password:	••••••••••••••••	<ul style="list-style-type: none"> Minimum 8 characters in length Contains 3/4 of the following items: <ul style="list-style-type: none"> Uppercase Letters Lowercase Letters Numbers Symbols 	
Hide:	<input checked="" type="checkbox"/>		
Score:	100%		
Complexity:	Very Strong		

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	$+(n^4)$	24	+ 96
Uppercase Letters	Cond/ Incr	$+(len-n)^2$	3	+ 42
Lowercase Letters	Cond/ Incr	$+(len-n)^2$	10	+ 28
Numbers	Cond	$+(n^4)$	7	+ 28
Symbols	Flat	$+(n^6)$	3	+ 18
Middle Numbers or Symbols	Flat	$+(n^2)$	9	+ 18
Requirements	Flat	$+(n^2)$	5	+ 10

Deductions	Type	Rate	Count	Bonus
Letters Only	Flat	-n	0	0
Numbers Only	Flat	-n	0	0
Repeat Characters (Case Insensitive)	Comp	-	12	- 3
Consecutive Uppercase Letters	Flat	$-(n^2)$	0	0
Consecutive Lowercase Letters	Flat	$-(n^2)$	6	- 12
Consecutive Numbers	Flat	$-(n^2)$	3	- 6
Sequential Letters (3+)	Flat	$-(n^3)$	0	0
Sequential Numbers (3+)	Flat	$-(n^3)$	0	0
Sequential Symbols (3+)	Flat	$-(n^3)$	0	0

Conclusion :

Creating and evaluating different passwords showed that length and character variety dramatically increase resistance to cracking. Local tools (cracklib-check) and entropy estimates corroborate online scores. Following passphrase-based best practices and using password managers significantly improves account security.