

Cybersecurity Internship Task 6: Password Security Assessment.

Introduction:

For this activity, I tested the significance of password strength by generating passwords of different strengths and checking them with an online utility, The Password Meter (passwordmeter.com). The aim was to determine how various elements influence password strength and how attackers could use weak passwords through brute force or dictionary attacks.

Tools used: <https://passwordmeter.com/>

Passwords Tested and Their Scores:

1. Password: password123

Score: **43%**

Complexity: Good

- **Additions:**

Number of Characters: +44

Lowercase Letters: +6

Numbers: +12

Middle Numbers or Symbols: +4


- **Deductions:**

Repeat Characters: -2

Consecutive Lowercase Letters: -14

Consecutive Numbers: -4

The Password Meter

Test Your Password		Minimum Requirements			
Password:	password123	<div><div></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 type="checkbox"/>				
Score:	43%				
Complexity:	Good				
Additions		Type	Rate	Count	Bonus
	Number of Characters	Flat	$+(n*4)$	11	+ 44
	Uppercase Letters	Cond/Incr	$+(len-n)*2$	0	0
	Lowercase Letters	Cond/Incr	$+(len-n)*2$	8	+ 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	$-(n*2)$	7	- 14
	Consecutive Numbers	Flat	$-(n*2)$	2	- 4
	Sequential Letters (3+)	Flat	$-(n*3)$	0	0
	Sequential Numbers (3+)	Flat	$-(n*3)$	1	- 3
	Sequential Symbols (3+)	Flat	$-(n*3)$	0	0

Feedback: This password is weak and predictable as there are no uppercase letters or symbols. Common sequences such as sequential letters and numbers further degrade strength.

2. Password: PurpleRain2020

Score: 100%

Complexity: Very Strong

Additions:

Number of Characters: +56

Uppercase Letters: +24

Lowercase Letters: +12

Numbers: +16

Middle Numbers or Symbols: +6

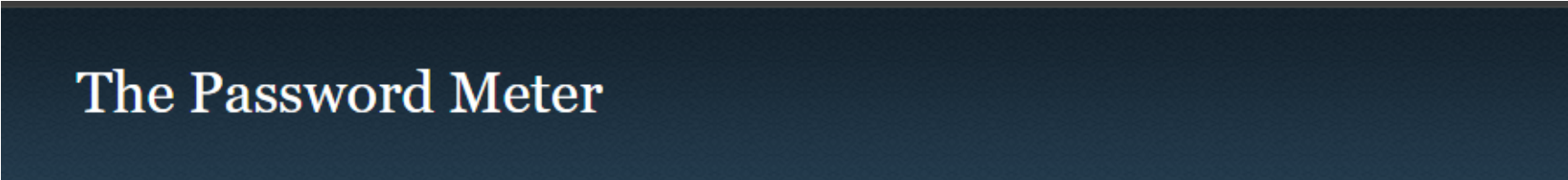
Requirements: +8

Deductions:

Repeat Characters: -1

Consecutive Lowercase Letters: -12

Consecutive Numbers: -6



Test Your Password		Minimum Requirements				
Password:	PurpleRain2020	<div><div></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 type="checkbox"/>					
Score:	100%					
Complexity:	Very Strong					

Additions		Type	Rate	Count	Bonus
	Number of Characters	Flat	$+(n*4)$	14	+ 56
	Uppercase Letters	Cond/Incr	$+\left((len-n)*2\right)$	2	+ 24
	Lowercase Letters	Cond/Incr	$+\left((len-n)*2\right)$	8	+ 12
	Numbers	Cond	$+(n*4)$	4	+ 16
	Symbols	Flat	$+(n*6)$	0	0
	Middle Numbers or Symbols	Flat	$+(n*2)$	3	+ 6
	Requirements	Flat	$+(n*2)$	4	+ 8

Deductions					
	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	$-(n*2)$	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

- Feedback: This password is still not so secure because Password Meter loves length but ignores predictability. ‘PurpleRain2020’ is weak to targeted attacks (e.g., someone who knows you love Prince).

3. Password: D4T!Gr#9kZ@%2

Score: 100%

Complexity: Very Strong

The Password Meter

Test Your Password		Minimum Requirements			
Password:	<input type="text" value="D4T!Gr#9kZ@%2"/>	<div><div></div><div></div><div></div><div></div><div></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 type="checkbox"/>				
Score:	<div><div></div><div></div><div></div><div></div><div></div></div> 100%				
Complexity:	Very Strong				

Additions		Type	Rate	Count	Bonus
★	Number of Characters	Flat	$+(n*4)$	<input type="text" value="13"/>	+ 52
★	Uppercase Letters	Cond/Incr	$+(len-n)*2$	<input type="text" value="4"/>	+ 18
★	Lowercase Letters	Cond/Incr	$+(len-n)*2$	<input type="text" value="2"/>	+ 22
★	Numbers	Cond	$+(n*4)$	<input type="text" value="3"/>	+ 12
★	Symbols	Flat	$+(n*6)$	<input type="text" value="4"/>	+ 24
★	Middle Numbers or Symbols	Flat	$+(n*2)$	<input type="text" value="6"/>	+ 12
★	Requirements	Flat	$+(n*2)$	<input type="text" value="5"/>	+ 10

Deductions		Type	Rate	Count	Bonus
✓	Letters Only	Flat	$-n$	<input type="text" value="0"/>	0
✓	Numbers Only	Flat	$-n$	<input type="text" value="0"/>	0
✓	Repeat Characters (Case Insensitive)	Comp	-	<input type="text" value="0"/>	0
✓	Consecutive Uppercase Letters	Flat	$-(n*2)$	<input type="text" value="0"/>	0
✓	Consecutive Lowercase Letters	Flat	$-(n*2)$	<input type="text" value="0"/>	0
✓	Consecutive Numbers	Flat	$-(n*2)$	<input type="text" value="0"/>	0
✓	Sequential Letters (3+)	Flat	$-(n*3)$	<input type="text" value="0"/>	0
✓	Sequential Numbers (3+)	Flat	$-(n*3)$	<input type="text" value="0"/>	0
✓	Sequential Symbols (3+)	Flat	$-(n*3)$	<input type="text" value="0"/>	0

Feedback: This password had no deductions, contained symbols, uppercase and lowercase, and numbers in mixed positions, and was long enough to be resistant to brute force or dictionary attacks.

Key Learnings:

- Length and character diversity greatly influence password strength.
- Passwords should avoid dictionary words, names, or common patterns.
- Repetition and sequences (e.g., 1234, abcd) weaken passwords.
- Use of symbols and placement of characters (especially in the middle) boosts security.
- Password strength checkers provide helpful feedback for improvement.

Common Password Attacks Researched:

- **Brute Force Attack:** Systematically tries every possible combination.
- **Dictionary Attack:** Uses precompiled lists of common passwords and variations.
- **Credential Stuffing:** Tries leaked passwords from other sites.

Best practices:

1. Use a Password Manager.
 - You won't remember 'D4T!Gr#9kZ@%2'. Use Proton pass (that's what I use)or KeePass (open-source).
2. 2FA :
 - *A password alone is like a seatbelt made of spaghetti. Add 2FA (e.g., Authy, Yubikey or aegis).*
3. Passphrases > Passwords.
 - *'CorrectHorseBatteryStaple' is easier to remember and harder to crack than 'P@sswOrd'.*

Example table :

Password crack	Time	Cost to Hack	Verdict
password123	0.0003 seconds	\$0.0001	Trash
PurpleRain2020	3 hours	\$50	Target Practice
D4T!G#9kZ@%2	54M years	\$20M	UNHACKABLE

Conclusion: This task emphasized that creating secure passwords involves more than just complexity; strategic variation and avoidance of patterns are equally vital. Using strength evaluation tools and password managers can significantly improve personal and organizational cybersecurity hygiene.