

Gerando Senhas Mais Fortes

Mário Leite

...

É do conhecimento de todos que a Informação é a moeda mais importante atualmente; na verdade, esse tipo de moeda é que dá proteção à moeda física no mundo moderno, afinal das contas. O tipo de proteção mais importante é a **criptografia**, que altera o texto da mensagem original através de substituição e/ou transposição de caracteres. O termo “criptografia” é originário do grego (*kryptós*=escondido + *gráphen*=escrita), e é definido pela enciclopédia Wikipédia como o “*estudo dos princípios e técnicas pelas quais a informação pode ser transformada da sua forma original para outra ilegível, de forma que possa ser conhecida apenas por seu destinatário*”. Na prática o que se deseja é transformar um texto legível em algo ilegível. O objetivo final da criptografia é dar segurança a um texto confidencial, tornando a informação nele contida inacessível para aquelas pessoas não autorizadas a conhecê-la.

Para as pessoas e para as empresas é importante usar senhas criptografadas para dificultar, ou mesmo impossibilitar, o acesso a seus dados/informações. Então, utilizar senhas bem “elaboradas” é fundamental para que o “curioso” seja desencorajado a tentar quebrar (*decryptar*) a senha. Muitas pessoas inventam suas próprias senhas de diversas maneiras; mas o perigo é criar uma senha muito óbvia e muito fácil de se descoberta, como o famoso e fraco “123” ou mesmo datas, que podem ser quebradas muito facilmente com métodos de “força bruta”. Por outro lado, existem rotinas que geram senhas bem mais difíceis de serem quebradas: são os “geradores de senhas”. Mas, o problema é saber se tais senhas são “fortes” (difíceis de serem quebradas) ou “fracas” (fáceis de serem quebradas). Um algoritmo muito popular para gerar senhas é usar a randomização para gerar caracteres pseudoaleatórios e usar uma sequência de tamanho desejado para gerar senhas. Entretanto, essa randomização pode ser “percebida” por um bom programador para descobrir essa sequência, quebrando a senha gerada. Mas existem bibliotecas próprias para gerar senhas bem mais fortes e sem usar randomização; é o caso da linguagem Python, que possui o pacote “**secrets**” que juntamente com “**string**” geram sequências não randômicas que podem ser utilizadas como senhas bem fortes e mais difíceis de serem quebradas. O programa “**CriptoSenhas2**” é um exemplo que gera senhas mais fortes que podem ser utilizadas em várias situações que precisam de mais proteção. A **figura 1** mostra um exemplo de senhas fortes geradas com o uso dessas duas bibliotecas.

```
Run: C:\Users\Usuario\PycharmProjects\pythonProject6\venv\Scripts\python.exe "D:\Livros\Livro11
Entre com o tamanho da senha [min: 8]: 30
Digite o número de senhas a gerar: 10
Nova senha gerada: X()J<^bGD`VP+9<i1H:+{"^tZ*A@"t
Nova senha gerada: nysHEW_7y}8'!1(T0sj'E0>wGd]cPE
Nova senha gerada: nZE!8Lnv&/i/BVNsx;V$`<\X^9?,,;e
Nova senha gerada: yUV^)"5ax1=U;Yx2.nX)1RQoVs?)oU
Nova senha gerada: (EIguI@zAJ3w*?!3FyoU}{*?j\}`8i
Nova senha gerada: -^:#}zI&6[;oh!n[`n4Bcigr0$^~cF
Nova senha gerada: p<DrEde/|y<3z0*]:yy,Fso;Q~qa}`
Nova senha gerada: zM|\]n<mIaoThe{c6QRa_XWa$IJCIf
Nova senha gerada: "DD;|@)1TZSt|xPn*Jht{xbF~dw3Q?
Nova senha gerada: -q)ttP1H`c&#FRK-3#::;[F+>-{^+Qz

Process finished with exit code 0
```

Figura 1 - Saída do programa “CriptoSenhas2”

```
'''
CriptoSenhas2.py
-----

Gera senhas mais fortes sem randomização para garantir segurança cibernética,
sem randomização..
-----

'''
import secrets
import string

def GerarSenha(tamanho):
    caracteres = string.ascii_letters + string.digits + string.punctuation
    senha = ''.join(secrets.choice(caracteres) for _ in range(tamanho))
    return senha

#=====
#Programa principal
if(__name__ == "__main__"):
    tam = 7
    while(tam < 8):
        tam = abs(int(input("Entre com o tamanho da senha [min: 8]: ")))

    numSenhas = abs(int(input("Digite o número de senhas a gerar: ")))
    for _ in range(numSenhas):
        novaSenha = GerarSenha(tam)
        print("Nova senha gerada:", novaSenha)
#Fim do programa "CriptoSenhas2" -----
```