Alexandre Andrioli Tucci

João Victor Saboya Ribeiro de Carvalho

Relatório Desempenho de Hash Criptográfico

1. Foi implementado Salt juntamente com o Hash já existente e o aumento de caracteres dentro da senha, é efetivo pois mesmo que um atacante tente adivinhar senhas, ele precisa calcular o Hash para cada tentativa usando o Salt específico de cada usuário. O que torna o processo muito mais demorado e computacionalmente custoso.

2. Resultados da Força Bruta, Secao1.py

Usuário	Hash	Senha	Tempo de quebra (segundos)
joao	d1c4e0fb6335e0e6787e74bf825359c2dfa6361f52e01445fb63f74016124264	S3n@	32.552
alex	e0ea3605f6c0cb5f0fcbbe54c65bdf2c8190667cd81d5b0164607296acbc30d1	aeio	0.040
mike	19252a7a3ab00e6c2dac91c45e4d0cb1fa71337dc66503f503229419756e651d	M#9a	27.864
hash	59d937723564937cd1441370c7c7cd9f669f93253dcfab8c50affad3b18d645e	ab12	0.017

Resultados da Força Bruta, Secao3.py

Usuário	Hash	Senha	Tempo de quebra (segundos)
joao	a0bb508a862af4cf98f7e117338c25836cb6906b41b032625306d3c7987d4443	P@ssw0rd	> 1500
alex	5a6d251b384ba49f11ec33c1410d3b480ef6569b60bf5f946ec4f5bb3cffe57d	alex123	> 1500
mike	637966b30fd7670e7d034219ce884d7e9803a2c428dab8473135af4968a413d5	F0rt3	> 1500
hash	20f0f4cd200b5ef1aea94c0a058f25cb43d4a8bd5cc219535ef460cd511f2944	S3nh@321	> 1500

A variação acontece porque cada senha é protegida com um Salt aleatório, o que torna cada tentativa de ataque única e mais demorada, já que o atacante precisa recalcular o Hash para cada combinação de senha e Salt.