Relatório – Desempenho de Hash Criptográfico

Arthur de Oliveira Carvalho e Leonardo Stall

1. Estratégias adotadas para dificultar a quebra de senhas

Na primeira implementação (secao1a.py), o sistema permitia que os usuários criassem nomes de usuário e senhas com apenas 4 caracteres. Esse cenário é bastante vulnerável a ataques de força bruta, pois o espaço de busca de combinações é pequeno.

Já na segunda implementação (secao1b.py), foi adotada uma estratégia de endurecimento da política de senha. O sistema passou a exigir senhas e nomes de usuários com no mínimo 8 caracteres. Essa exigência aumenta exponencialmente o espaço de busca e o tempo necessário para quebrar as senhas, o que torna ataques de força bruta muito mais difíceis e lentos.

Essa mudança é efetiva contra ataques de força bruta pois:

- Aumenta o número de combinações possíveis: ao passar de 4 para 8 caracteres, mesmo com o mesmo conjunto de caracteres (letras minúsculas e números), o número de combinações cresce de 36^4 (≈1,6 milhão) para 36^8 (≈2,8 trilhões).
- Obriga o usuário a usar padrões mais robustos, dificultando tentativas simples baseadas em dicionários ou nomes comuns.
- Aumenta o custo computacional do atacante, tornando o ataque inviável em muitos cenários.

2. Análise de hash criptográfico

Para realizar a quebra dos hashes, foram utilizados dois códigos Python: secao2a.py e secao2b.py. Ambos implementam um algoritmo de força bruta para verificar todas as combinações possíveis de senhas, comparando o hash SHA-256 de cada tentativa com os hashes armazenados nos arquivos JSON (base_usuarios2a.json e base_usuarios2b.json).

Abaixo, apresentamos os resultados obtidos:

Tabela 1 – Resultados da quebra de senha para base_usuarios2a.json (senha com 4 caracteres)

Usuário	Hash	Senha	Tempo de quebra (segundos)
artt	3c083577cea84c7a71466df2b953 9ba2c55acee27cc8da2da59fcaac8c158f54	artt	0.12
attr	2148952c2c47033e44ec61fa83f01 d8688db767553a0d2b6260428373d938fb8	attr	0.12
atrr	ce59d3f3f85755add573be55652c3 88479600aac9b03ddca53571a9bf934317f	atrr	0.10
rtta	b8b7c05a082bc069f222d4d999bbd9 feee7a02c100962d12d87602a0a375401e	rtta	1.36

Tempo total: 1.7

Tabela 2 – Resultados da quebra de senha para base_usuarios2b.json (senha com 8 caracteres)

Usuário	Hash (16 primeiros caracteres)	Senha	Tempo de quebra (segundos)
Arthur05	(não quebrada)	—	—
arthur05	(não quebrada)	—	—
aRthur05	(não quebrada)	—	—
arThur05	(não quebrada)	—	—

Observações:

- O código de força bruta executado sobre os dados de base_usuarios2b.json ainda não conseguiu quebrar nenhuma das senhas. Isso ocorre devido à complexidade significativamente maior imposta pelo comprimento mínimo de 8 caracteres.

Análise da variação dos tempos de quebra

A diferença de desempenho entre as duas seções (Tabela 1 e Tabela 2) está diretamente relacionada à complexidade das senhas. Enquanto as senhas de 4 caracteres são rapidamente quebradas por força bruta (em frações de segundo), as senhas de 8 caracteres aumentam drasticamente o tempo de execução, tornando o ataque impraticável para execução simples.

Essa variação ilustra bem a importância de políticas de senha mais seguras, que aumentam o custo computacional dos ataques e protegem melhor os dados dos usuários.

Essa estratégia mostra-se eficaz especialmente em sistemas onde a autenticação é um ponto crítico, sendo altamente recomendada a obrigatoriedade de senhas com tamanho mínimo, combinação de letras, números e, se possível, símbolos.