



1. Função detectar_invasao

Essa função é responsável por analisar os registros de log e identificar um possível invasor com base em critérios específicos.

Parâmetro:

- registros: Lista de strings, onde cada string representa um registro no formato "id_usuario:status" [por exemplo, "user1:falha" ou "user2:sucesso"].

Lógica:

1. Inicialização de variáveis:

- usuario_atual: Armazena o ID do usuário atualmente sendo processado.
- tentativas_consecutivas: Conta as falhas consecutivas de um usuário.
- invasor_detectado: Armazena o ID do invasor quando identificado.

2. Iteração sobre os registros:

- Para cada registro no formato id_usuario:status:
 - O registro é dividido em id_usuario [identificador do usuário] e status [pode ser "sucesso" ou "falha"].

3. Verificação de continuidade:

- Se o id_usuario for igual ao último usuário processado:
 - Se o status for "falha", incremente o contador de falhas consecutivas.
 - Se o contador ultrapassar 3, o usuário é marcado como invasor, e o loop termina.
 - Se o status for "sucesso", o contador de falhas consecutivas é resetado.
- Se o id_usuario for diferente:
 - Verifica se o usuário anterior teve mais de 3 falhas consecutivas. Se sim, ele é considerado invasor, e o loop é encerrado.
 - O usuario_atual é atualizado para o novo ID, e o contador é iniciado de acordo com o status do novo registro.

4. Checagem final:

- Após o loop, verifica se o último usuário processado teve mais de 3 falhas consecutivas.

5. Resultado:

- Retorna o ID do invasor, caso detectado, ou uma mensagem indicando que nenhum invasor foi encontrado.

2. Função main

Essa função gerencia a interação com o usuário e organiza o fluxo do programa.

Fluxo:

- Solicita ao usuário que insira os registros de log como uma única string, com registros separados por vírgulas [ex.: "user1:falha, user1:falha, user1:falha, user1:sucesso"].
- Processa a entrada, criando uma lista de registros limpos:
 - Remove espaços em branco extras e aspas.
- Chama a função detectar_invasao com os registros.
- Exibe o resultado retornado.

3. Estrutura de execução

- O programa é projetado para ser executado diretamente.
- Quando executado, a função main() é chamada.

Exemplo de uso:

Entrada:

"user1:falha, user1:falha, user1:falha, user1:falha, user2:sucesso"

Saída:

user1

Explicação: O usuário user1 teve 4 falhas consecutivas, o que indica uma tentativa suspeita de invasão.

Pontos principais:

- O programa considera mais de 3 falhas consecutivas como comportamento suspeito.
- Reseta o contador de falhas ao mudar de usuário ou após uma tentativa bem-sucedida.
- Finaliza a análise assim que um invasor é detectado, otimizando o desempenho.