

# Algoritmo para atualizações de arquivos em Python

## Descrição do Projeto

Este projeto descreve um algoritmo em Python criado para automatizar a atualização de uma "allow list" (lista de permissões) de endereços IP. O script lê um arquivo de texto contendo os IPs autorizados, remove endereços específicos que estão em uma "lista de remoção" e, por fim, salva o arquivo com a lista atualizada. Esta é uma tarefa de manutenção comum em segurança cibernética para garantir que apenas IPs confiáveis tenham acesso a um sistema.

## Abra o arquivo que contém a lista de permissões

Para começar, usei o `with open()` para abrir o arquivo em modo de leitura `"r"`. Isso garante que o arquivo seja fechado automaticamente após a leitura, evitando erros.

Eu atribuí o `allow_list.txt` como uma string para a variável `import_file`:

```
# Assign `import_file` to the name of the file  
import_file = "allow_list.txt"
```

Depois utilizei o `with` para abrir o arquivo:

```
# First line of `with` statement  
with open(import_file, "r") as file:
```

## Leia o conteúdo do arquivo

Com o arquivo aberto, utilizei o método `.read()` para ler todo o conteúdo do arquivo e armazená-lo como uma única string na variável `ip_addresses`.

```
with open(import_file, "r") as file:
    # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`
    ip_addresses = file.read()

# Display `ip_addresses`
print(ip_addresses)
```

```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

## Converta a String para uma Lista

Aqui utilizei o método `.split()` para separar as strings (nesse caso, IPs) com base nos espaços entre eles, em formato de lista.

```
# Use `.split()` to convert `ip_addresses` from a string to a list
ip_addresses = ip_addresses.split()

# Display `ip_addresses`
print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.168.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

## Iterar pela lista de remoção

Utilizei o laço de repetição `for` para percorrer cada endereço IP (`element`) que foi lido do arquivo `allow_list.txt`.

```
for element in ip_addresses:
    # Build conditional statement
    # If current element is in `remove_list`,
```

## Remover endereços IPs que estão na lista de remoção

Dentro do laço, vai ter uma condição `if` que verifica se o endereço IP atual (`element`) está presente na `remove_list`. Se estiver presente, o método `.remove()` vai ser usado para excluir ele da lista `ip_addresses`.

```
for element in ip_addresses:
    # Build conditional statement
    # If current element is in `remove_list`,
    if element in remove_list:
        # then current element should be removed from `ip_addresses`
        ip_addresses.remove(element)
```

## Atualizar o arquivo com a lista revisada de endereços IPs

Aqui a lista de IPs atualizada é convertida de volta para uma string com o método `.join()`, com cada IP separado por um espaço. Em seguida, o arquivo original é aberto novamente, mas desta vez em modo de escrita "w", que apaga o conteúdo antigo. Por fim, o método `.write()` escreve a nova string no arquivo.

```
# Convert `ip_addresses` back to a string so that it can be written into the text file  
ip_addresses = " ".join(ip_addresses)  
  
# Build `with` statement to rewrite the original file  
with open(import_file, "w") as file:  
    # Rewrite the file, replacing its contents with `ip_addresses`  
    file.write(ip_addresses)
```

## Sumário

Eu criei um algoritmo para remover endereços IP de um arquivo de permissões chamado `allow_list.txt`. O processo envolveu ler o arquivo, converter seu conteúdo em uma lista de IPs e, em seguida, **iterar sobre essa lista**. Para cada IP da lista de permissões, o código verificava se ele também existia na `remove_list`. Se existisse, era removido com o método `.remove()`. No final, a lista atualizada foi convertida de volta para uma string com o método `.join()` e usada para sobrescrever o arquivo original, completando a atualização.