

Algorithm for file updates in Python

Project description

Mi rol en la empresa es de un analista de seguridad que debe actualizar regularmente el archivo que identifica a los empleados que pueden acceder a contenido restringido. Para esto tengo una lista de direcciones IP permitidas para identificar a los usuarios activos y una lista de direcciones IP que debo eliminar para sacarles el acceso.

Open the file that contains the allow list

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

Read the file contents

```
# Use `file.read()` to read the imported file and store it in a variable named `ip_addresses`  
ip_addresses = file.read()
```

Convert the string into a list

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

Iterate through the remove list and Remove IP addresses that are on the remove list

```
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)
```

Update the file with the revised list of IP addresses

```
# 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)
```

Summary

Para comenzar, utilicé la sentencia **with** y la función **open()** para abrir el archivo de texto y leí su contenido mediante el método **.read()**. A continuación, transformé esa cadena de texto en una lista utilizando el método **.split()**, lo cual me permitió gestionar cada dirección IP de forma individual. Posteriormente, iteré a través de esta lista con un bucle **for** y apliqué un condicional **if** para identificar y eliminar las direcciones prohibidas usando **.remove()**. Finalmente, convertí la lista revisada nuevamente en un formato de texto con el método **.join()** y sobrescribí el archivo original utilizando **.write()** para guardar los cambios.