Algorithm for file updates in Python

Project description

You are a security professional working at a health care company. As part of your job, you're required to regularly update a file that identifies the employees who can access restricted content. The contents of the file are based on who is working with personal patient records. Employees are restricted access based on their IP address. There is an allow list for IP addresses permitted to sign into the restricted subnetwork. There's also a remove list that identifies which employees you must remove from this allow list.

Your task is to create an algorithm that uses Python code to check whether the allow list contains any IP addresses identified on the remove list. If so, you should remove those IP addresses from the file containing the allow list.

Open the file that contains the allow list

Firstly, I imported “allow_list.txt” and assigned it to the import_file value. Then using command “with open(import_file, “r”) as file:” I opened the import_file. Then I assigned it to ip_addresses variable.

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

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
```

Convert the string into a list

Using this command:

```
ip_addresses = ip_addresses.split()
```

I converted IP addresses that were saved as a string into a list. It makes removing IP addresses easier.

Iterate through the remove list

Now I had to iterate through the remove list to choose the IP addresses to be removed.

```
for element in remove_list:

    # Create conditional statement to evaluate if `element` is in `ip_addresses`

    if element in ip_addresses:

        # use the `.remove()` method to remove
        # elements from `ip_addresses`

        ip_addresses.remove(element)
```

In every iteration using condition statement "if element in ip_addresses:" I check whether an element from "remove_list" occurs in IP addresses. If the condition is met it removes this element from the IP addresses list.

Update the file with the revised list of IP addresses

```
ip_addresses = "\n".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)

# Build `with` statement to read in the updated file

with open(import_file, "r") as file:

    # Read in the updated file and store the contents in `text`

    text = file.read()
```

After changes to IP addresses made I needed to convert this list back to the string. I achieved that by using command: "ip_addresses = "n".join(ip_addresses)". Then using command "with open(import_file, "w") as file:" I rewritten the import_file with the updated contents of

“ip_addresses”. At the end I assigned the readable format of “import_file” to variable text.

Summary

I created an algorithm that removes IP addresses identified in a remove_list variable from the "allow_list.txt" file of approved IP addresses. This algorithm involved opening the file, converting it to a string to be read, and then converting this string to a list stored in the variable ip_addresses. I then iterated through the IP addresses in remove_list. With each iteration, I evaluated if the element was part of the ip_addresses list. If it was, I applied the .remove() method to it to remove the element from ip_addresses.. After this, I used the .join() method to convert the ip_addresses back into a string so that I could write over the contents of the "allow_list.txt" file with the revised list of IP addresses.