

Algorithm for file updates in Python

Project description

In our organization, we manage access to restricted content through an allow list of specified IP addresses, identified in the "allow_list.txt" file. To ensure security, we also maintain a separate remove list that identifies IP addresses no longer authorized to access the content. I've developed an algorithm to automate the process of updating the "allow_list.txt" file, efficiently removing IP addresses that should no longer have access. This automated approach streamlines the management of authorized access and enhances security by promptly revoking access for addresses on the remove list.

Open the file that contains the allow list

For the first part of the algorithm, I opened the "allow_list.txt" file. First, I assigned this file name as a string to the import_file variable:

```
In [2]: import_file = "allow_list.txt"
```

Then, I used a with statement to open the file:

```
In [6]: import_file = "allow_list.txt"
with open(import_file, "r") as file:
```

Read the file contents

In order to read the file contents, I used the .read() method to convert it into the string.

```
In [13]: import_file = "allow_list.txt"
with open(import_file, "r") as file:
    ip_addresses = file.read()
    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
```

Convert the string into a list

In order to remove individual IP addresses from the allow list, I needed it to be in list format. Therefore, I next used the `.split()` method to convert the `ip_addresses` string into a list:

```
In [15]: import_file = "allow_list.txt"

with open(import_file, "r") as file:
    ip_addresses = file.read()

    ip_addresses = ip_addresses.split()

    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']
```

Iterate through the remove list

A key part of my algorithm involves iterating through the IP addresses that are elements in the `remove_list`. To do this, I incorporated a for loop:

```
In [28]: import_file = "allow_list.txt"

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

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

    ip_addresses = file.read()
    ip_addresses = ip_addresses.split()

    for element in ip_addresses:
        print(element)

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
```

Active
Go to S

Remove IP addresses that are on the remove list

My algorithm requires removing any IP address from the allow list, `ip_addresses`, that is also contained in `remove_list`. Because there were not any duplicates in `ip_addresses`, I was able to use the following code to do this:

In [29]:

```
import_file = "allow_list.txt"

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

with open(import_file, "r") as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split()

for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
print(ip_addresses)

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

Update the file with the revised list of IP addresses

As a final step in my algorithm, I needed to update the allow list file with the revised list of IP addresses. To do so, I first needed to convert the list back into a string. I used the `.join()` method for this:

In [37]:

```
import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split()

for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
ip_addresses = " ".join(ip_addresses)

with open(import_file, "w") as file:
    file.write(ip_addresses)

print(ip_addresses)

ip_address 192.168.25.60 192.168.205.12 192.168.6.9 192.168.52.90 192.168.90.124 192.168.186.176 192.168.133.188 192.168.203.198 192.168.218.219 192.168.52.37 192.168.156.224 192.168.60.153 192.168.69.116
```

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.