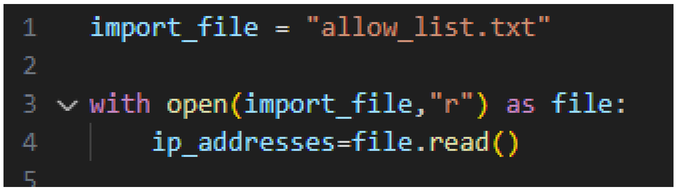
Algorithm for file updates in Python

Project description

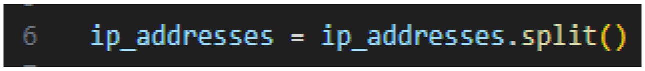
At my organisation, access to restricted content is controlled with a list of known IP addresses. The "allow_list.txt" file identifies these IP addresses. A separate remove list identifies IP addresses that should not have access to this content. I created an algorithm to automate updating the "allow_list.txt" file and removing these IP addresses that should no longer have access.

Open the allow list file and convert contents to a list



```python
import_file = "allow_list.txt"

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



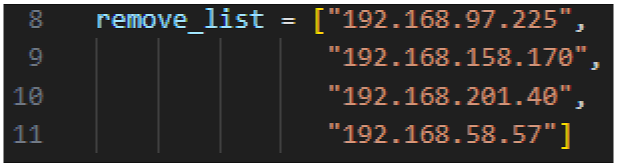
```python
ip_addresses = ip_addresses.split()
```

I opened the file using a with statement. The open function allows the user to open a file and store the contents in a variable. In the screenshot above, I used the “r” specifier so that I could read the contents of the “allow_list.txt” file.

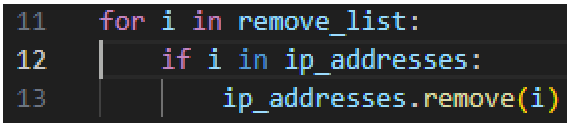
The contents of the file were then required to be stored in a variable for further processing. To do this, I used the built-in .read() function. This converts the file contents to a string.

I then used the .split() function to convert the “ip_addresses” string to a list. Given the immutable nature of strings, it is necessary to reassign the return value of the split function.

Iterate through the remove list and remove unwanted IPs



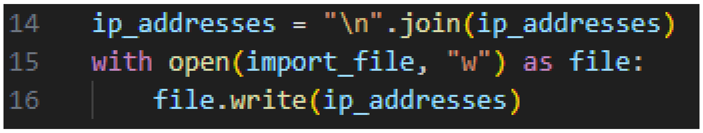
```python
remove_list = ["192.168.97.225",
               "192.168.158.170",
               "192.168.201.40",
               "192.168.58.57"]
```



```python
for i in remove_list:
    if i in ip_addresses:
        ip_addresses.remove(i)
```

To iterate through the remove list I used a for loop. I used a conditional statement to compare each element of the remove list with that of the allow list. When an element from the remove list appears in the allow list, the condition is met and the IP address is removed from the allow list using the .remove() function.

Update the file with the revised list of IP addresses



```
ip_addresses = "\n".join(ip_addresses)
with open(import_file, "w") as file:
    file.write(ip_addresses)
```

When all iterations have completed, the allow list is ready to be updated. The list needs to be converted back to the form that it was received at the beginning. To do this, I used the join function to join all elements of the list back to a string. The “\n” tells python to separate each list entry by a new line character. Using the with statement, I can open the file again and using “w” allows me write access to the file. I write the updated list of allowed IPs back to the file.

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.