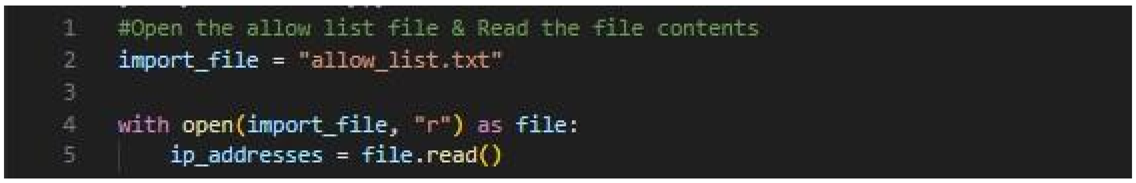
Algorithm for file updates in Python

Project description

I created a Python script that automatically updates a list of allowed IP addresses for a restricted network. The script reads the list from a file, removes any IPs that are on a separate removal list, and writes the updated list back to the file. This project shows my skills in handling files, using loops and conditionals, and working with strings and lists to automate tasks that support cybersecurity operations.Open the file that contains the allow list

Read the file contents

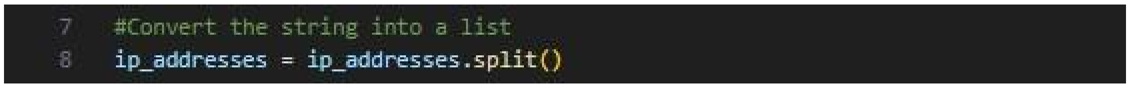


```python
#Open the allow list file & Read the file contents
import_file = "allow_list.txt"

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

We use the open() function in read mode ("r") inside a with statement to safely access the allow_list.txt file. The .read() method reads all the file contents at once into a single string, stored in the variable ip_addresses. The with statement ensures the file is properly closed after reading, even if an error occurs.

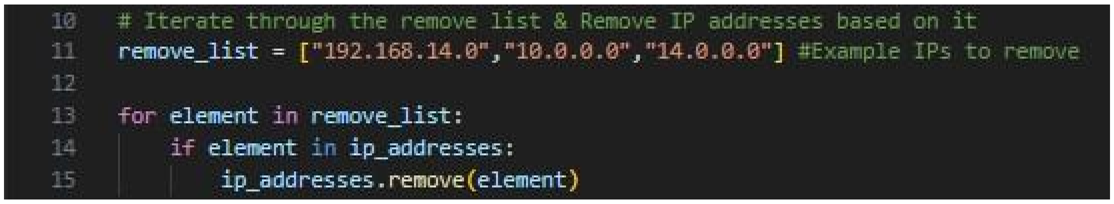
Convert the string into a list



```python
#Convert the string into a list
ip_addresses = ip_addresses.split()
```

The .split() method is applied to the ip_addresses string to break it into a list of individual IP addresses. By default, .split() separates the string at whitespace characters like spaces or newlines. This makes it easier to work with each IP address as a separate element in a Python list.

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

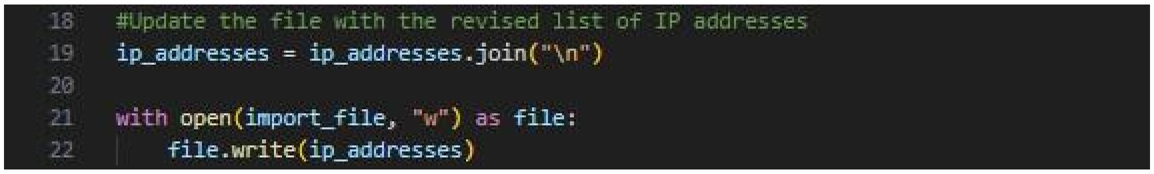


```python
# Iterate through the remove list & Remove IP addresses based on it
remove_list = ["192.168.14.0","10.0.0.0","14.0.0.0"] #Example IPs to remove

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

A for loop is used to go through each IP address in the remove_list. Inside the loop, we use an if statement to check if the current IP exists in the ip_addresses list. If it does, the .remove() method deletes it. This ensures that only unwanted IP addresses are removed without affecting others. This works safely because there are no duplicate IP addresses.

Update the file with the revised list of IP addresses



```python
#Update the file with the revised list of IP addresses
ip_addresses = ip_addresses.join("\n")

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

We use the .join() method with "\n" to convert the updated list back into a clean string, placing each IP address on a new line. Then, using open() in write mode ("w") inside another with statement, we overwrite the original file with the revised content using the .write() method.

Summary

In this project, a Python algorithm was developed to automate updating an IP allow list. First, I opened and read the file contents safely using a with statement and .read(). Then, I converted the data into a list using .split() to easily manipulate each IP. A for loop iterated through the remove_list, and the .remove() method was applied to delete matching IPs. Finally, I rebuilt the updated list into a string with .join() and overwrote the original file using .write(). This ensures that the allow list remains up-to-date and only includes authorized IP addresses.