Algorithm for file updates in Python

Project description

This project describes a Python algorithm created to automate the updating of an IP address "allow list". The script reads a text file containing authorized IPs, removes specific addresses found in a "remove list," and saves the file with the updated list. This is a common cybersecurity maintenance task performed to ensure that only trusted IPs have access to a system.

Open the file that contains the allow list

The `with open()` statement is used to open the file in read mode `"r"`. This method ensures that the file is automatically closed after its contents have been read, which helps prevent errors.

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

import_file = "allow_list.txt"
```

```
# First line of `with` statement

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

Read the file contents

With the file open, the `.read()` method is used to read the entire content of the file and store it as a single string in the `ip_addresses` variable.

```
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
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

The `.split()` method is used here to parse the string of IPs into a list, using the spaces between them as delimiters. This conversion allows for the individual manipulation of each IP address.

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

ip_addresses = ip_addresses.split()

# 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', '192.16
8.90.124', '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 `for` loop is used to iterate through each IP address (`element`) that was read from the `allow_list.txt` file.

```
for element in ip_addresses:

  # Build conditional statement
  # If current element is in `remove_list`,
```

Remove IP addresses that are on the remove list

Inside the loop, an `if` condition checks if the current IP address (`element`) is present in the `remove_list`. If it is, the `.remove()` method is used to delete it from the `ip_addresses` 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

The updated list of IPs is converted back into a string using the `.join()` method, with each IP address separated by a space. Then, the original file is opened again, this time in write mode `"w"`, which erases the old content. Finally, the `.write()` method writes the new string to the file.

```
# 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

I created an algorithm to remove IP addresses from a permissions file named `allow_list.txt`. The process involved reading the file, converting its content into a list of IPs, and then iterating through that list. For each IP on the allow list, the code checked if it also existed in the `remove_list`. If it did, it was removed using the `.remove()` method. In the end, the updated list was converted back to a string using the `.join()` method and was used to overwrite the original file, completing the update.