Algorithm for file updates in Python

Project description

In my organization, access to restricted resources is controlled through an IP-based allow list, stored in a file named allow_list.txt.
Over time, certain IP must be revoked due to policy update, detected threats or access termination. Those are stored in a separated list, called remove_list.
To reduce manual error and improved efficiency, I developed a Python script that automatically: Reads the current allow list, Remove unauthorized IP addresses, Update the file.
This automation support secure access control management and reduce risk of outdated permission.

Open the file that contains the allow list

For the first part, I opened up the "allow_list_txt" file. Assign a variable name for it.

```
import_file = "allow_list.txt"
```

Then i use the with statement for open the file:

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

The with statements ensures proper resource management by automatically closing the file after execution. "r" argument specifies that file is opened in read mode.

Read the file contents

To read the content of the file, I used the .read() method:

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

The .read() method converts the file content into a string. The result is stored in the variable ip_addresses.

Convert the string into a list

To manipulate individual IP, i converted the string into a list, using the .split() method:

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

By default the .split() command separated elements by whitespaces. Now each IP address is an individual element in the list. This allows a safe and structured removal of specific entries.

Iterate through the remove list

Next, I iterated through the remove_list using for loops:

```
for element in remove_list:
```

The loop allows the script to check each IP address that needs to be revoked.

Remove IP addresses that are on the remove list

Inside the loop, I added a conditional check to prevent errors:

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

The if condition ensure that the IP exists in the allow list before attempting removal. Then .remove() method deletes the matching IP address from the list. This step ensure that only authorized IP remain the allow list.

Update the file with the revised list of IP addresses

After removing the necessary IP, I converted the list back into a properly formatted string:

```
ip_addresses = "\n".join(ip_addresses)
```

The \n separator ensure that each IP is written on a new line.
Reopened the file in write mode and update its contents:

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

The “w” mode overwrites the file with the updated allow list.

Summary

In this project, I created aPython script that automates the maintenance of an IP allow list. The scripts do : Open and read allow list file; Converts its content into a list; iterates through a removal list; Deletes unauthorized IP addresses; Updates the file with a revised data.
This approach improves access control hygiene, reduce manual administrative work,and help maintain secure network resource management