Update allowed IP address file with Python

Project description

The health care company is using an allow list of IP addresses file to determine which IP's are allowed to access content and which ones are not allowed. This allow list is stored in a file called "allow_list.txt". The following is the process used to update the allow list file to remove IP addresses that should no longer be able to access the restricted content.

Open the file that contains the allow list

The first step is to open the current "allow_list.txt" file to read the contents of current IP addresses allowed to access the restricted content. This is done by first setting the name of the file "allow_list.txt" to the variable import_file within the update_file function which can be seen below:

```
# Call `update_file()` and pass in "allow_list.txt" and a list of IP addresses to be removed

update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])
```

Once the file is set to the variable import_file it is opened with python and the file is read which can be see in the following method:

```
def update_file(import_file, remove_list):

  # Build `with` statement to read in the initial contents of the file

  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()
```

Read the file contents

The .read() method is used to extract the contents of the file into a string variable called ip_addresses:

```
  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()
```

Convert the string into a list

The .split() method is used to convert the string variable into a list of strings.

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

ip_addresses = ip_addresses.split()
```

By converting the string into a list of strings will make it easier to iterate through the list so we can find the IP addresses that need to be removed.

Iterate through the remove list

Once we have the list we can iterate through the list by using a for loop. If an element in the list matches with an element in the remove list that element will be removed from the ip_addresses list which can be seen below:

```
# Build iterative statement
# Name loop variable `element`
# Loop through `ip_addresses`

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)
```

Remove IP addresses that are on the remove list

We use the conditional if statement to identify if the element matches in the remove_list variable:

```
  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

Once the iteration is complete the .join method is used to convert the list of strings back into one big string.

```
# Convert `ip_addresses` back to a string so that it can be written into the text file

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

Then the python open() method is used to write to the existing “allow_list.txt” file.

```
# 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

This algorithm is used to remove IP addresses that are identified in the remove_list variable that might exist with the “allow_list.txt” file. It does this by opening the file and iterating over the values to identify the IP addresses. Below is the full method used to do this process

```
# Define a function named `update_file` that takes in two parameters: `import_file` and `remove_list`
# and combines the steps you've written in this lab leading up to this

def update_file(import_file, remove_list):

  # Build `with` statement to read in the initial contents of the file

  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()

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

  ip_addresses = ip_addresses.split()

  # Build iterative statement
  # Name loop variable `element`
  # Loop through `ip_addresses`

  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)

  # 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)

# Call `update_file()` and pass in "allow_list.txt" and a list of IP addresses to be removed

update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])

# Build `with` statement to read in the updated file

with open("allow_list.txt", "r") as file:

  # Read in the updated file and store the contents in `text`

  text = file.read()

# Display the contents of `text`

print(text)
```