🛡️ Python Algorithm: Updating IP Allow List for Secure Access by Jordan Butler

🔍 Project Description

As a security professional at a healthcare company, it's critical to manage access to sensitive data and systems. This project demonstrates the use of Python to automate the process of updating an IP allow list. The algorithm checks if any IP addresses on a removal list are present in the allow list and removes them if found. This activity reflects real-world cybersecurity practices such as access control, automation, and secure file handling.

📂 File Structure and Purpose

- `allow_list.txt`: Contains the current list of allowed IP addresses.
- `remove_list.txt`: Contains IP addresses that should be removed from the allow list.
- `update_allow_list.py`: Python script to perform the update.

🧠 Python Algorithm

📌 Step 1: Open and Read the Allow List

```
with open("allow_list.txt", "r") as file:
    allow_list_data = file.read()
```

- **Explanation**:
  - `with` ensures the file is automatically closed after use (resource management).
  - `open("filename", "r")` opens the file in **read mode**.
  - `.read()` reads the entire contents of the file as a string.

📌 Step 2: Convert String to a List

```
allow_list = allow_list_data.splitlines()
```

- **Explanation**:
  - `.splitlines()` splits the string at line breaks to create a list of IP addresses.

📌 Step 3: Read and Parse the Remove List

```
with open("remove_list.txt", "r") as file:
    remove_list = file.read().splitlines()
```

- **Explanation**:
    - Reads the file and uses `.splitlines()` to produce a list of IPs to remove.

📌 Step 4: Remove IPs from the Allow List

```
for ip in remove_list:
    if ip in allow_list:
        allow_list.remove(ip)
```

- **Explanation**:
    - `for` loop iterates through each IP in the remove list.
    - `if ip in allow_list` checks if the IP is present.
    - `.remove(ip)` deletes the IP from the list.

📌 Step 5: Write the Updated List Back to File

```
with open("allow_list.txt", "w") as file:
    for ip in allow_list:
        file.write(ip + "\n")
```

- **Explanation**:
    - File is opened in **write mode** (`"w"`), which overwrites its content.
    - Loop writes each IP followed by a newline to keep formatting consistent.
    - `.write()` is used to output each IP to the file.

✅ Summary

This project demonstrates basic file I/O operations in Python and shows how cybersecurity professionals can automate access control updates. The script:

- Opens and reads files securely with `with` and `open()`.
- Parses file content using `.read()` and `.splitlines()`.
- Iterates over data with `for` loops.

- Modifies lists using `.remove()`.
- Writes updated data back using `.write()`.

🧪 Sample Files (for testing)

allow_list.txt

```
192.168.1.1
10.0.0.5
172.16.0.2
```

remove_list.txt

```
10.0.0.5
```

Output after running the script

```
192.168.1.1
172.16.0.2
```

📝 Self-Assessment Checklist (✅ = complete)

1. ✅ I created a Python script to open, read, and write files.
2. ✅ I used the `with` statement and `open()` function correctly.
3. ✅ I used `.read()` and `.write()` methods in the algorithm.
4. ✅ I used the `.split()` or `.splitlines()` method to parse strings.
5. ✅ I used a `for` loop to iterate through the remove list.
6. ✅ I used the `.remove()` method to update the list.
7. ✅ I provided code screenshots or typed code.
8. ✅ I wrote detailed explanations of the code syntax and logic.
9. ✅ I included a summary and project description.