# **Python Algorithm for File Updates**

### **Project description**

As a security professional for a fictional healthcare company, I developed a Python algorithm to update a file containing IP addresses authorized to access restricted patient data. The algorithm removes IP addresses listed in a separate remove list from the "allow\_list.txt" file, ensuring only authorized users maintain network access.

### **Open the file that contains the allow list**

I assigned the filename "allow\_list.txt" to the variable import\_file. Then, using a with statement combined with the open() function in read mode ("r"), I opened the file safely for reading. The with statement ensures the file closes automatically after reading.



I then used a with statement to open the file.



### **Read the file contents**

I used the .read() method to load the entire content of the allow list file into a string variable ip\_addresses.



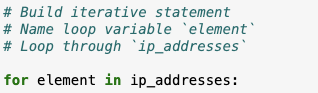
### **Convert the string into a list**

Next, I converted the string of IP addresses into a list using the .split() method. This splits the string by whitespace into individual IP addresses for easier manipulation.

****

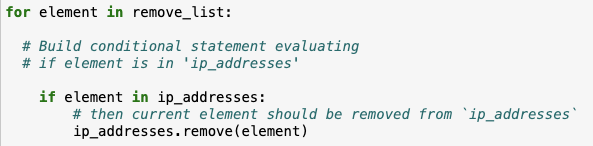
### **Iterate through the remove list**

Iterating through IP addresses listed in the remove\_list was achieved through a for loop. This loop iterated over the sequence of IP addresses stored in the ip\_addresses list, applying specific operations to each element.



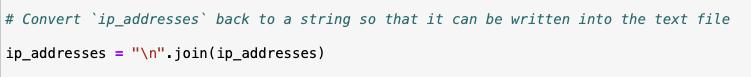
### **Remove IP addresses that are on the remove list**

Inside the loop, I checked if the IP was present in the ip\_addresses list and removed it using .remove() if found.

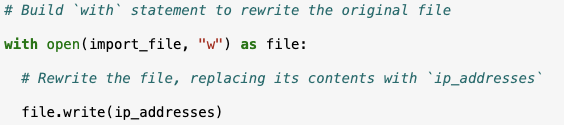


### **Update the file with the revised list of IP addresses**

To finalize the algorithm, updating the allow list file with the revised list of IP addresses was necessary. Firstly, the list had to be converted back into a string format, accomplished using the .join() method, which concatenates elements of an iterable into a single string with specified separators.



Employing .join() on the ip\_addresses list allowed me to create a string with each IP address on a separate line (using the string "\n" as a separator), ideal for updating the "allow\_list.txt" file.



The with statement and .write() method facilitated this update, with the "w" argument indicating the intention to overwrite the file contents. By appending .write() to the file object within the with statement, I replaced the file's previous content with the updated ip\_addresses data, thereby ensuring restricted content accessibility only to approved IP addresses.

### **Summary**

This Python algorithm automates the update of an IP allow list by reading from a file, converting the contents to a list, removing specified IPs, and writing the updated list back to the file. It maintains secure network access control by ensuring only authorized IPs remain in the allow list.