# Algorithm for file updates in Python

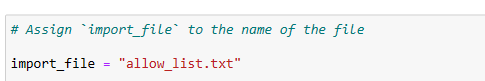
## Project description

As a security professional working at my organization, I was required to regularly update a file that identifies employees who can access restricted content. The contents of the file are based on who is working with personal patient information. Employees are restricted access based on their ip address. There is an allow list "allow\_list.txt" for ip addresses permitted to sign into the restricted subnetwork. There is also a remove list that identifies which employees I should remove from this allow list.

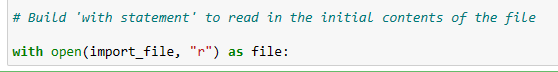
As a security analyst, I'm responsible for developing an algorithm that parses a file containing ip addresses that are allowed to access restricted content and remove addresses that no longer have access.

## Open the file that contains the allow list

For the first part of the algorithm, I opened the text file “allow\_list.txt” and assigned it using the import\_file variable.



Then, I used the with statement to open the file.



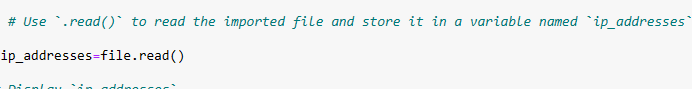
In this project, the with statement is used with the .open() function in read mode to open the allow list file for the purpose of reading it. The purpose of opening the file is to allow me to access the IP addresses stored in the allow list file. The withstatement allows me to efficiently handle files and will help manage the resources by closing the file after exiting the with statement.

The open() function in Python allows me to open a file. At the first parameter, it takes in the name of the file( or a variable containing the name of the file). As the second parameter, it takes in a string that indicates how the file should be handled.

Pass in the letter“r”as the second parameter when you want to read the file.

I also used the as keyword to assign a variable name file; file stores the output of the open() function.

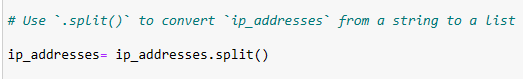
## Read the file contents



The .read() method in Python allows you to read in a file, convert it to a string and store it in a variable named ‘ip\_addresses’. I called file.read()to read the imported file.

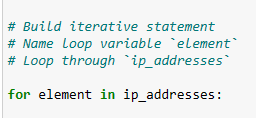
So, this code reads the contents of the "allow\_list.txt" file into a string format that allows me to later use the string to organize and extract data in my Python program.

## Convert the string into a list



In order to remove individual ip addresses from the allow list, the ip addresses need to be in a list format. Therefore, I used a split()method to convert the ip\_addresses string into a list. To store this list, I reassigned it back to the variable ip\_addresses. The purpose of splitting ip\_addresses into a list is to make it easier to remove IP addresses from the allow list. By default, the .split() function splits the text by whitespace into list elements.

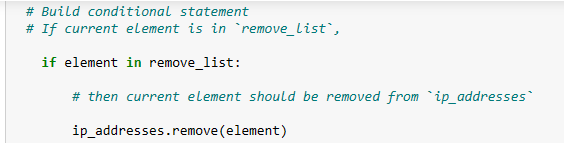
## Iterate through the remove list



In order to remove the elements of remove\_list from the ip\_addresses, I’ll need both an iterative statement and a conditional statement. The for loop in Python repeats code for a specified sequence. The overall purpose of the for loop in a Python algorithm like this is to apply specific code statements to all elements in a sequence.

First, I built a for loop to iterate through ip\_addresses. I named the loop variable element, l and used In as a loop condition. The keyword in indicates to iterate through the sequence ip\_addresses and assign each value to the loop variable element.

## Remove IP addresses that are on the remove list



Now,I built a conditional statement to remove the elements of a remove\_list from the ip\_addresses list.I used the inoperator to check if the element is in remove\_list. To remove elements from ip\_addresses, I call the .remove()method on ip\_addresses, and pass in the element. In every iteration, if the current element in the ip\_addresses list is in the remove\_list, the remove method should be used to remove that element.

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

## 

Now that I've removed these Ip addresses from the ip\_addresses variable, I need to complete the algorithm by updating a file with this revised list. To do this, I must first convert the ip\_addresses list back to a string using the .join()method.

To complete the first line of the with statement, I call the open() function and pass in the name of the file as the first parameter “import\_file” and the letter “with” as the second parameter.

Inside the with statement, I call the .write()method to replace the content of the file with the data stored in ip\_addresses. Inside the with statement, I call file.write()and pass in ip\_addresses.

## Summary

I created an algorithm that removes IP addresses from the "allow\_list.txt"file of approved IP addresses that are listed in a remove\_list variable. This algorithm involves opening the file, converting it to a string that could be read, and then turning that string into a list that was stored in the variableip\_addresses. The IP addresses in remove\_list were then iterated through. With each iteration,I determined whether an element was a part of the ip\_addresses list. If it was, I removed the element from ip\_addresses by applying the.remove()method to it. I then used the.join() function to turn the IP addresses back into strings so that I could replace the content of the IP address list in the "allow\_list.txt" file with the revised list of ip addresses.