# Algorithm for file updates in Python

## Project description

As an aspiring security analyst, I am exploring how Python programming language can be utilized for automation of some task. As a practice, I worked on a text file containing IP addresses that are allowed to access specific restricted content in an organization. One of the task of a security analyst is to control the access to restricted content.

In this project, I parsed the text file with a filename of allow\_list.txt to read and update the contents of it. I developed an algorithm that parses the text file that contains IP addresses, and updates the content by removing the addresses that no longer have an access to the restricted content.

## Open the file that contains the allow list

I declared a variable named import\_file with a string value that contains the file name of the text file.

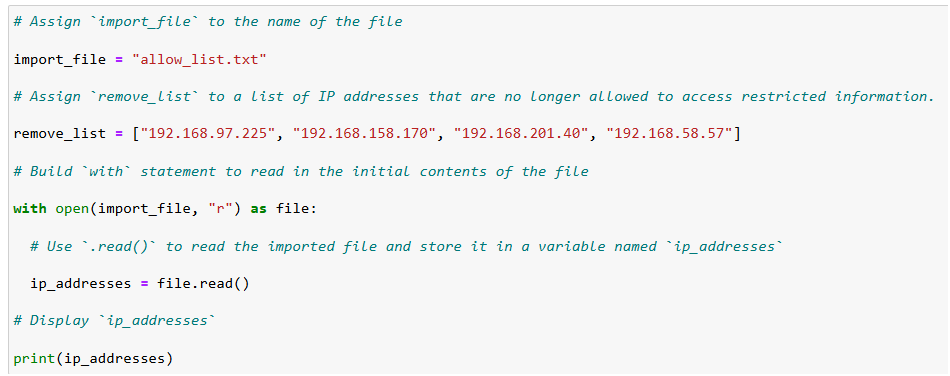


To open the file, I used the with statement with an “r” parameter to indicate that the file contents will be read.



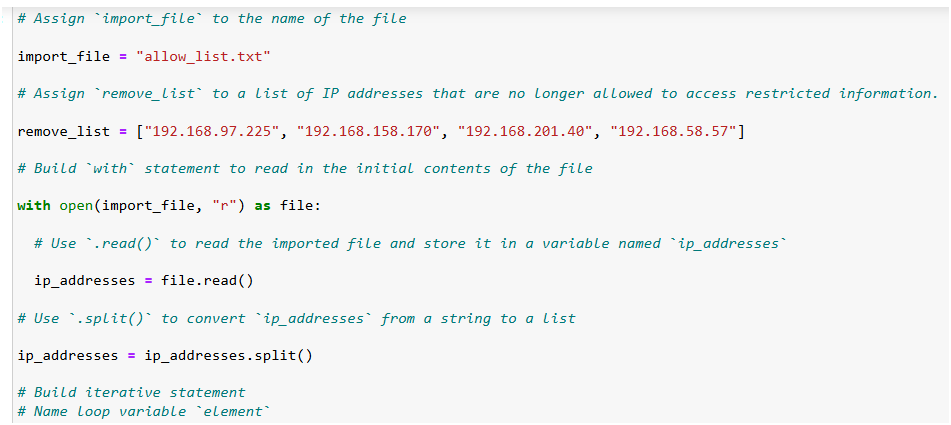
## Read the file contents

I converted the text file contents into a string through the use of file.read(). The converted text file is stored in the ip\_addresses.



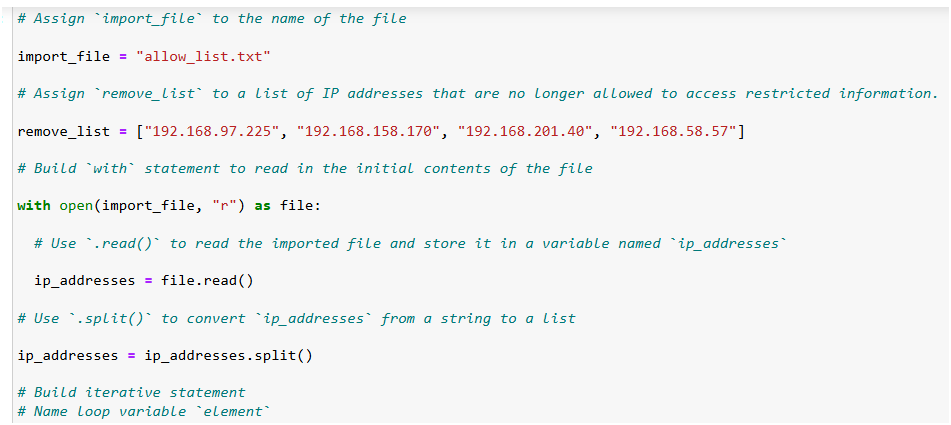
## Convert the string into a list

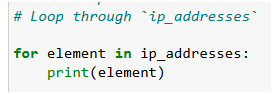
To convert the string content to a list, I used split() method.



## Iterate through the remove list

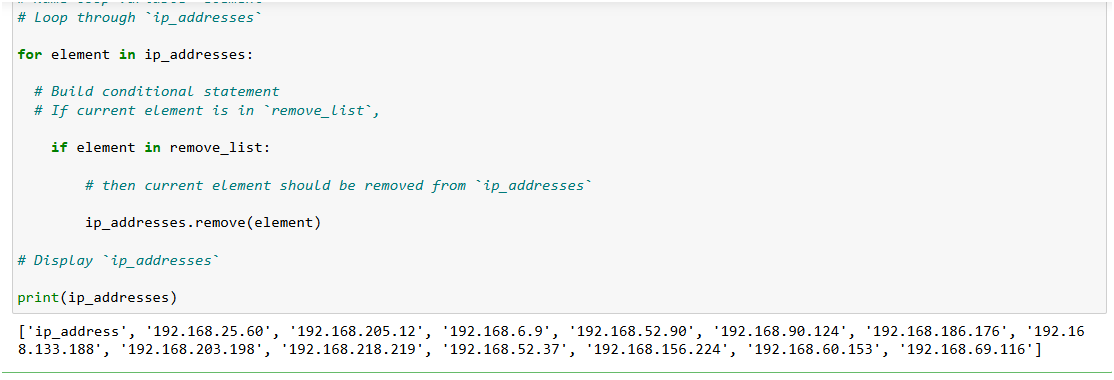
To analyze the contents of remove\_list, I created an algorithm that will iterate over the contents of the remove\_list array through the use of a for loop.





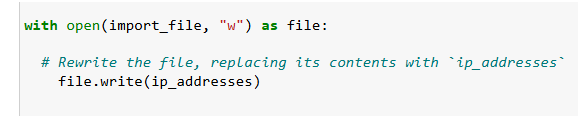
## Remove IP addresses that are on the remove list

This algorithm is created to determine if the ip\_addresses coming in belongs to the remove\_list. If it is included in the remove\_list, it will be removed through the remove() method.



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

This algorithm is used to open and update the file.



## Summary

To remove the IP addresses that belongs to the remove\_list, I created an algorithm to read, write, and update the allow.list text file. The processes include opening the file, converting the text file contents into a string type to make it readable, convert the string to a list stored in a declared variable, iterate over the elements to evaluate if the element is part of the list where there are two conditions to determine if the .remove() method will be utilized, .used .join() method to convert the ip\_addresses back into a string, and updated the imported file. The source code used is included below.

*Final Source Code:*

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### YOUR CODE HERE ###

# 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)