# Algorithm for file updates in Python

## Project description

I am working as a security professional for a health care company, I am tasked to create an algorithm that uses Python code to check whether the allow list contains any IP addresses identified on the remove list. If so, you should remove those IP addresses from the file containing the allow list.

## Open the file that contains the allow list

First I opened the “allow\_list.txt” file, then I assigned the file name as a string called import\_file.



Then using the with statement, I was able to open the file as expected.

A white background with black text

Description automatically generated

In my algorithm, the with statement is used with the .open() function in read mode to open

the allow list file for the purpose of reading it. I am opening the file to allow me to

access the IP addresses stored in the allow list file. In the code with

open(import\_file, "r") as file:, the open() function has two parameters. The first

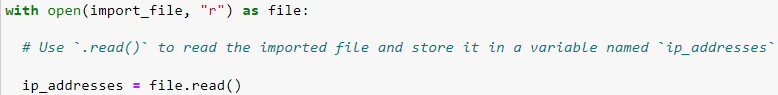
identifies the file to import, and then the second indicates what I want to do with the file. In this

case, "r" indicates that I want to read it. The code also uses the as keyword to assign a

variable named file; file stores the output of the .open() function while I work within the

with statement.

## Read the file contents



The .open() function included the argument "r" for “read,” enables me to call the

.read() function in the body of the with statement, .read() converts the file

into a string and allows it to be read. I applied the .read() method to the file variable

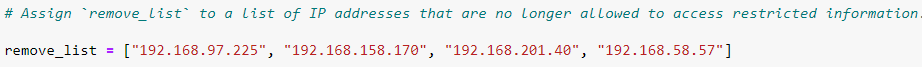
identified in the with statement. Then, I assigned the string output of this method to the

variable ip\_addresses.

This is the result for my argument

A close up of a text

Description automatically generated



A screenshot of a computer

Description automatically generated

The above result does contain the removed\_list along with the allow\_list.txt, we need to use another algorithm command that allows the list to be updated to change this result.

## Convert the string into a list

## 

The .split() function is called by appending it to a string variable. It works by converting the contents of a string to a list. 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. In this algorithm, the .split() function takes the data stored in the variable ip\_addresses, which is a string of IP addresses that are each separated by a whitespace, and it converts this string into a list of IP addresses. To store this list, I reassigned it back to the variable ip\_addresses.



## Iterate through the remove list

A screenshot of a computer program

Description automatically generated

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.

## Remove IP addresses that are on the remove list

A computer code with black text

Description automatically generated

First, within the for loop, I created a conditional that evaluated if the element was found in the ip\_addresses list or not. Then I added the .remove() to elements so that if they were not found in ip\_addresses it would result in an error, I passed in the loop variable element as the argument so that each IP address that was in the remove\_list would be removed from ip\_addresses.

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

## A white background with black text Description automatically generated

The .join() method combines all items in an iterable into a string. The .join() method is

applied to a string containing characters that will separate the elements in the iterable once

joined into a string.

## A screenshot of a computer code Description automatically generated

This time, I used a second argument of "w" with the open() function in the with statement.

This argument indicates that I want to open a file to write over its contents. When using this

argument "w", I can call the .write() function in the body of the with statement. The

.write() function writes string data to a specified file and replaces any existing file content.

## Summary

I created an algorithm that removes IP addresses that identified in a remove\_list variable from

the "allow\_list.txt" file of approved IP addresses. This algorithm includes opening the

file, converting it to a string to be read, and then converting this string to a list stored in the

variable ip\_addresses. I then iterated through the IP addresses in remove\_list. With each

iteration, I evaluated if the element was part of the ip\_addresses list. If it was, I applied the

.remove() method to it to remove the element from ip\_addresses.. After this, I used the

.join() method to convert the ip\_addresses back into a string so that I could write over

the contents of the "allow\_list.txt" file with the revised list of IP addresses.