# Algorithm for file updates in Python

## Project description

In this project, the objective is to develop an algorithm in Python for updating a file containing an allowed list of IP addresses. The project aims to enhance cybersecurity measures by developing a Python algorithm for managing access to restricted content. The algorithm focuses on parsing a text file containing an allowed list of IP addresses, removing specified IP addresses from this list, and updating the file with the revised set of IP addresses.

## Open the file that contains the allow list

The first step involves opening the file that contains the list of allowed IP addresses. This file, named "allow\_list.txt," serves as the reference for controlling access to specific restricted content within the organization.

## Read the file contents

Following the file opening, the algorithm reads the contents of "allow\_list.txt" using the read() function. This step captures the initial set of IP addresses stored in the file as a single string, providing the foundation for subsequent processing.

## Convert the string into a list

To facilitate manipulation and analysis, the algorithm employs the split() function to convert the string of IP addresses into a list. The resulting list, named "ip\_addresses," becomes a mutable structure suitable for iterative operations.

## Iterate through the remove list

The algorithm incorporates a loop to iterate through each element in the "ip\_addresses" list. During this iteration, a conditional statement checks whether the current element is present in the specified "remove\_list.txt." If a match is found, the algorithm proceeds to remove the IP address from the "ip\_addresses" list.

## Remove IP addresses that are on the remove list

For each matching IP address found in the "remove\_list.txt," the algorithm removes the corresponding element from the "ip\_addresses" list. This step ensures that IP addresses no longer authorized to access restricted content are excluded from the updated list.

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

After removing unauthorized IP addresses, the algorithm rejoins the modified "ip\_addresses" list into a string format using the join() function. Subsequently, the algorithm reopens the original "allow\_list.txt" file in write mode and replaces its content with the revised list of IP addresses. This step ensures that the file is updated with the latest set of authorized IP addresses.

## Summary

In summary, the developed algorithm automates the process of updating the allowed list of IP addresses. By parsing the file, removing unauthorized IP addresses, and updating the file with the revised list, the algorithm contributes to maintaining a secure environment by controlling access to restricted content. The modular approach of the algorithm allows for flexibility and ease of integration into broader cybersecurity frameworks.

| # Define a function named `update\_file` that takes in two parameters: `import\_file` and `remove\_list` # and combines the steps you've written in this lab leading up to this  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("allow\_list.txt", "remove\_list.txt")  # 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) |
| --- |