

# Algorithm for file updates in Python

## Project description

A security professional at a healthcare organization is responsible for maintaining secure access to sensitive patient data. As part of their duties, they routinely manage an allow list—a file containing IP addresses authorized to access a restricted subnetwork where personal patient records are stored.

To uphold access control policies and ensure only active, authorized personnel retain access, the security professional uses a remove list, which includes IP addresses of employees who no longer require access due to role changes, departure, or other factors.

The task involves creating a Python-based algorithm that automates this process. The algorithm checks the allow list for any IP addresses that appear in the remove list and removes them accordingly. This helps maintain data security and compliance with healthcare privacy standards by ensuring access is limited strictly to current, authorized users.

## Opening the file that contains the allow list

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# First line of `with` statement
with open(import_file, "r") as file:
```

## Reading the file contents

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# 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()

# Display `ip_addresses`
print(ip_addresses)
```

```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

## Converting the string into a list

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# 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()

# Display `ip_addresses`
print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.168.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

## Iterateing through the remove list

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a List of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# 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:

    # Display `element` in every iteration
    print(element)
```

```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

## Removing IP addresses that are on the remove list

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# 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 == remove_list:

        # then current element should be removed from `ip_addresses`
        ip_addresses.remove(element)

# Display `ip_addresses`
print(ip_addresses)

['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.90', '192.168.158.170', '192.168.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

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

```
# Assign `import_file` to the name of the file
import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# 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)
```

## Summary

In this activity, a Python-based solution was developed to manage access control for a restricted healthcare subnetwork. The task involved reading an "allow list" file containing IP addresses of authorized employees and removing any IP addresses listed in a separate "remove list." The solution was implemented using core Python concepts such as functions, file handling, loops, conditional statements, and list operations. A custom function was created to read the allow list, filter out IPs found in the remove list, and overwrite the original file with the updated list. This approach ensures that only authorized personnel maintain access to sensitive patient data, aligning with security and privacy requirements in the healthcare domain.