MY FIRST PYTHON WORK

June 17, 2024

Creation of another algorithm for a cyber security operation

0.1 Introduction

An important part of cybersecurity is controlling access to restricted content. In this project, I'll work with a text file containing IP addresses that are allowed to access specific restricted content at your organization.

Parsing a file allows security analysts to read and update the contents. Python helps analysts develop algorithms to automate the process of parsing files and keeping them up-to-date.

I am to develop an algorithm that parses this text file of IP addresses and updates the file by removing that addresses that no longer have access to the restricted content.

0.2 Scenario

In this project, I worked as a security analyst and I am responsible for developing an algorithm that parses a file containing IP addresses that are allowed to access restricted content and removes addresses that no longer have access.

0.3 Task 1

My goal is to develop an algorithm that parses a series of IP addresses that can access restricted information and removes the addresses that are no longer allowed.

I am given a text file called `"allow_list.txt"` that contains a series of IP addresses that are allowed to access restricted information.

There are IP addresses that should no longer have access to this information, and their IP addresses need to be removed from the text file. I am given a variable named `remove_list` that contains the list of IP addresses to be removed.

I'll display both variables to explore their contents, and run the cell.

[3]:
```
# 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"]

# Display `import_file`

print(import_file)

# Display `remove_list`

print(remove_list)
```

```
allow_list.txt
['192.168.97.225', '192.168.158.170', '192.168.201.40', '192.168.58.57']
```

Successful.

The contents of the output are: allow_list.txt [‘192.168.97.225’, ‘192.168.158.170’, ‘192.168.201.40’, ‘192.168.58.57’]**

allow_list.txt

[‘192.168.97.225’, ‘192.168.158.170’, ‘192.168.201.40’, ‘192.168.58.57’]

0.4 Task 2

In this task, I’ll assign ‘import_file’ to the name of the file.

[7]:
```
# 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:
```

```
  File "<ipython-input-7-4b1a57787374>", line 12

^
```

```
SyntaxError: unexpected EOF while parsing
```

Error is expected.

0.5 Task 3

Now, I'll use the `.read()` method to read the imported file and store it in a variable named `ip_addresses`.

Afterwards, I'll display `ip_addresses` to examine the data in its current format.

[24]:
```python
# 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("allow_list.txt", "r") as file:
    allow = file.read()

  # Use `.read()` to read the imported file and store it in a variable named␣
↪`ip_addresses`
ip_addresses = allow

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
```

SUCCESSFUL

[25]:
```python
# 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']
```

[26]:
```python
# 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(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']
['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']
['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']
['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']
```

```
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
['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']
```

[27]:
```
# 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)

# Display `ip_addresses`

print(ip_addresses)
```

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

[29]:
```
# 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)
```

[36]:
```
# 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)

# 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)
```

[ ]:

```
# 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 ### YOUR CODE HERE ###

    # 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)
```

[34]:
```
# Define a function named `update_file` that takes in two parameters:␣
 ↪`import_file` and `remove_list`
```

```python
# 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
```

```python
update_file("allow_list.txt", "ip_addresses")

# 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)
```

```
192.168.205.12 192.168.6.9 192.168.52.90 192.168.90.124 192.168.186.176
192.168.133.188 192.168.218.219 192.168.52.37 192.168.156.224 192.168.60.153
192.168.69.116
```

0.6 Conclusion

Python has functions and syntax that help you import and parse text files. - The `with` statement allows you to efficiently handle files. - The `open()` function allows you to import or open a file. It takes in the name of the file as the first parameter and a string that indicates the purpose of opening the file as the second parameter. - Specify `"r"` as the second parameter if you're opening the file for reading purposes. - Specify `"w"` as the second parameter if you're opening the file for writing purposes. - The `.read()` method allows you to read in a file. - The `.write()` method allows you to append or write to a file. - You can use a `for` loop to iterate over a list. - You can use an `if` statement to check if a given value is in a list and execute a specific action if so. - You can use the `.split()` method to convert a string to a list. - You can use Python to compare contents of a text file against elements of a list. - Algorithms can be incorporated into functions. When defining a function, you must specify the parameters it takes in and the actions it should execute.