**Introduction**

An important part of cybersecurity is controlling access to restricted content. In this lab, you'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.

You'll 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.

**Tips for completing this lab**

**Scenario**

In this lab, you're working as a security analyst and you're 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.

**Task 1**

Your eventual 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. Python can automate this process.

You're 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. You're given a variable named remove\_list that contains the list of IP addresses to be removed.

Display both variables to explore their contents, and run the cell. Be sure to replace each ### YOUR CODE HERE ### with your own code before running the following cell.

A screenshot of a computer program

Description automatically generated

By running this code, you'll see the filename of the allow list and the IP addresses to be removed printed on the console. This helps verify that the data is loaded correctly before proceeding with the removal process.

**Task 2**

In this task, start by opening the text file using the import\_file variable, the with keyword, and the open() function with the "r" parameter. Be sure to replace the ### YOUR CODE HERE ### with your own code.

For now, you'll write the first line of the with statement. Running this code will produce an error because it will only contain the first line of the with statement; you'll complete this with statement in the task after this.

A screenshot of a computer error

Description automatically generated

**Task 3**

Now, use the .read() method to read the imported file and store it in a variable named ip\_addresses.

Afterwards, display ip\_addresses to examine the data in its current format.

Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Finding matching IP addresses:**

Although the code doesn't explicitly compare the contents of allow\_list.txt with remove\_list, we can see from the printed output that there are matching IP addresses.

* IP addresses "192.168.158.170" and "192.168.58.57" are present in both the remove\_list and the output of the file contents.

**Task 4**

After reading the file, reassign the ip\_addresses variable so its data type is updated from a string to a list. Use the .split() method to achieve this. Adding this step will allow you to iterate through each of the IP addresses in the allow list instead of navigating a large string that contains all the addresses merged together.

Afterwards, display the ip\_addresses variable to verify that the update took place.

Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell.

A screenshot of a computer program

Description automatically generated

Breaking down the code:  **Import the filename:**

* The first line import\_file = "allow\_list.txt" assigns the filename containing the allowed IP addresses to a variable named import\_file.

 **Define the list to remove:**

* The second line defines a list named remove\_list containing four IP addresses. These IP addresses are intended to be removed from the allowed list.

 **Print statements:**

* The next two lines are print statements intended to show the filename containing the allowed IP addresses and the contents of the remove\_list.

 **Open the allowed list file:**

* The line with open (import\_file, "r") as file: opens the file named "allow\_list.txt" for reading in a secure manner using a with statement. This ensures that the file is closed properly even if errors occur.

 **Read the file contents:**

* Inside the with block, the line ip\_addresses = file.read() reads the contents of the file into a variable named ip\_addresses.

 **Convert file contents to a list:**

* The line ip\_addresses = ip\_addresses.split("\n") converts the string containing the IP addresses, read from the file, into a list where each element is a single IP address. This is done by splitting the string based on new line characters (\n).

 **Print the contents (optional):**

* The commented-out line # Print the contents of ip addresses (optional) suggests that the code might have originally included printing the contents of ip\_addresses but it's commented out now.

**Task 5**

* Now, you'll write code that removes the elements of remove\_list from the ip\_addresses list. This will require both an iterative statement and a conditional statement.
* First, build the iterative statement. Name the loop variable element, loop through ip\_addresses, and display each element. Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell.
* A screenshot of a computer program

  Description automatically generated

**Assign filename:**

*  The first line import\_file = "allow\_list.txt" assigns the filename containing the allowed IP addresses to a variable named import\_file.

 **Define list to remove:**

* The second line defines a list named remove\_list containing four IP addresses. These IP addresses are intended to be removed from the allowed list.

 **Print statements:**

* The next two lines are print statements that are used to display the filename containing the allowed IP addresses and the contents of the remove\_list.

 **Open the file for reading:**

* The line with open (import\_file, "r") as file: opens the file named "allow\_list.txt" for reading in a secure manner using a with statement. This ensures that the file is closed properly even if errors occur.

 **Read the file contents:**

* Inside the with block, the line ip\_addresses = file.read() reads the contents of the file into a variable named ip\_addresses.

 **Convert file contents to list:**

* The line ip\_addresses = ip\_addresses.split("\n") converts the string containing the IP addresses, read from the file, into a list where each element is a single IP address. This is done by splitting the string based on new line characters (\n).

 **Iterate through the list:**

* The for element in ip\_addresses: line starts a loop that iterates through each element in the ip\_addresses list.

 **Print each element (optional):**

* The line print(element) inside the loop prints each element (IP address) in the ip\_addresses list. This line is likely for debugging purposes and may not be part of the final functionality.

**Task 6**

* Now, build a conditional statement to remove the elements of remove\_list from the ip\_addresses list. The conditional statement should be placed inside the iterative statement that loops through ip\_addresses. In every iteration, if the current element in the ip\_addresses list is in the remove\_list, the remove() method should be used to remove that element.
* Afterwards, display the updated ip\_addresses list to verify that the elements of remove\_list are no longer in the ip\_addresses. Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell.

A screenshot of a computer program

Description automatically generated

The code snippet reads a list of allowed IP addresses from a file named "allow\_list.txt" and stores each IP address as a separate element in a list named ip\_addresses. It then defines a list of IP addresses to be removed from the allowed list but doesn't explicitly perform the removal. Finally, it prints the filename containing the allowed IP addresses and the contents of the remove\_list.

**Task 7**

The next step is to update the original file that was used to create the ip\_addresses list. A line of code containing the .join() method has been added to the code so that the file can be updated. This is necessary because ip\_addresses must be in string format when used inside the with statement to rewrite the file.

The .join() method takes in an iterable (such as a list) and concatenates every element of it into a string. The .join() method is applied to a string consisting of the character that will be used to separate every element in the iterable once its converted into a string. In the code below, the method is applied to the string " ", which contains just a space character. The argument of the .join() method is the iterable you want to convert, and in this case, that's ip\_addresses. As a result, it converts ip\_addresses from a list back into a string with a space between each element and the next.

After this line with the .join() method, build the with statement that rewrites the original file. Use the "w" parameter when calling the open() function to delete the contents in the original file and replace it with what you want to write. Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell. This code cell will not produce an output.

A screenshot of a computer program

Description automatically generated

Basically, the script reads a list, filters out unwanted entries, and writes the cleaned list back to the file. It’s a straightforward example of reading, processing, and writing data, which are common tasks in many programming applications, especially in fields like cybersecurity where managing access lists might be a frequent requirement. This kind of script shows how you can automate updates to data in a controlled way, reducing the chance for human error.

**Task 8**

In this task, you'll verify that the original file was rewritten using the correct list.

Write another with statement, this time to read in the updated file. Start by opening the file. Then read the file and store its contents in the text variable.

Afterwards, display the text variable to examine the result.

Be sure to replace each ### YOUR CODE HERE ### with your own code before you run the following cell.

Alright, let's go through this code like we're tackling our first intern project at a tech company.

We've got a Python script that's handling a list of IP addresses. Here's the breakdown of what each part is doing:

1. **Assign Filenames and IP Lists**:
   * We set up the name of our file where allowed IPs are stored in a variable called import\_file.
   * Then, we've got this remove\_list which has IPs that we need to yank from the allowed list.
2. **Print and Check**:
   * Just printing out the names of the file and IPs we plan to remove. Think of it like a sanity check before we dive in.
3. **Open and Read the Allow List**:
   * We open our allow\_list.txt file in read mode. That's what the 'r' is for.
   * Then, we grab all the IPs in there and split them by newlines. Now we've got a nice list of IPs, each one its own item in the list called ip\_addresses.
4. **The Removal Loop**:
   * Here's the workhorse part. We loop through all the IPs in ip\_addresses.
   * If any IP matches one from remove\_list, we remove it right there on the spot.
5. **Prepare for Saving**:
   * We take our cleaned-up list, ip\_addresses, and stitch it back into a string with newlines. This way, it's ready to go back into a file format.
6. **Write it Back**:
   * Now we open up allow\_list.txt again, this time in write mode ('w'). This mode lets us overwrite the file with our updated list.
   * We drop our updated IPs into the file, all neat and tidy.
7. **Verify Our Work** (Optional):
   * We're being thorough, so we open the file again to read the updated list.
   * Then we print it out to see our handiwork. It's like proofreading your essay before you hand it in.
8. **Final Print**:
   * The script prints the final, updated list of IPs to the console, so we can double-check everything went smoothly.

Summary:

This Python script is an automated tool for updating an allow list of IP addresses. It begins by defining the target file (allow\_list.txt) and a pre-determined list of IP addresses slated for removal. The script prints these initial parameters for user verification.

It proceeds to open the allow list file in read mode, reads the contents into memory, and parses the data into a list of individual IP addresses. Utilizing a for loop and a conditional statement, the script iteratively checks for and removes any IP addresses that are present in the removal list.

After the removal process, the script transforms the updated list back into a string format suitable for file storage, with each IP address separated by a newline character. The allow list file is then opened in write mode, and the updated list of IP addresses is written back, effectively overwriting the previous contents.

Optionally, for confirmation of the successful update, the script can reopen the allow list file in read mode and print the new contents to the console. This final step ensures the integrity of the update process and provides immediate feedback on the operation's outcome.