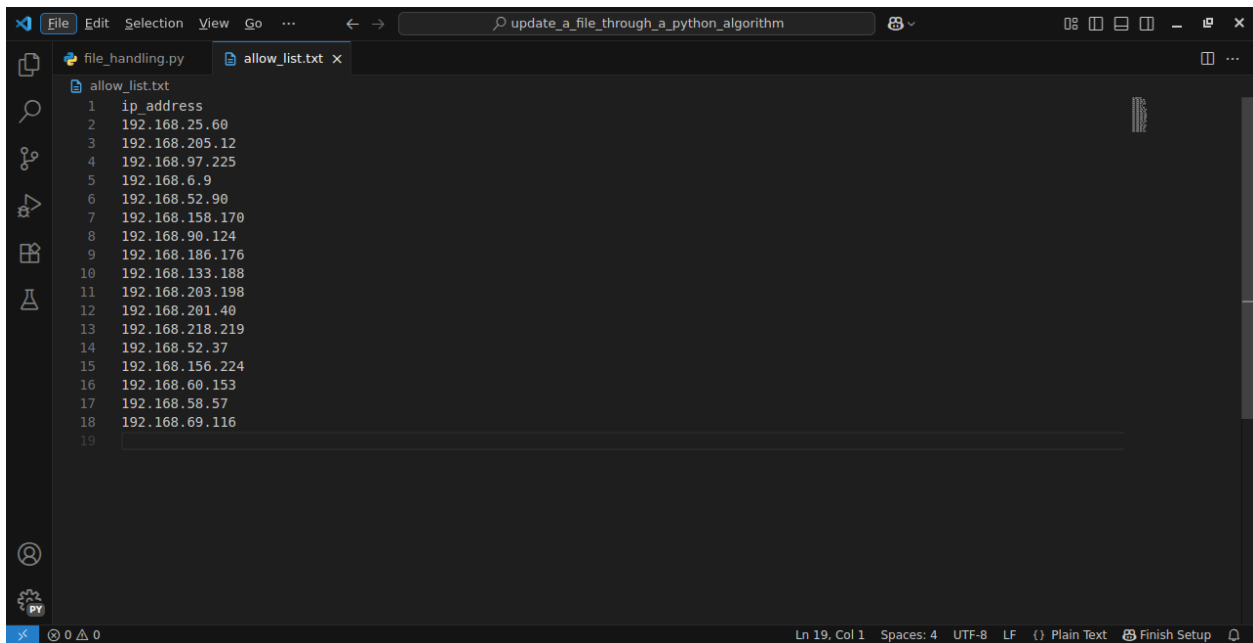


Algorithm for file updates in Python

Project description

This algorithm provides upgrade automation of files containing allowed IP addresses. The function needs to execute inside the same folder that needs to be upgraded. Provide the name of the file and a list of denied IPs and the upgrade in the file is automatically executed. `update_file(import_file, remove_list)` function is designed to efficiently manage IP allow lists. Users provide a file name (the application needs to be executed within that file's folder) and a list of denied or blocked IPs for removal. The algorithm reads the file, converts its content into a list, and then iterates through the provided `remove_list` to identify and remove unauthorized IP addresses. Finally, through robust file management using `with` and `open()`, the updated list safely overwrites the original file's contents, ensuring the list is current and secure."

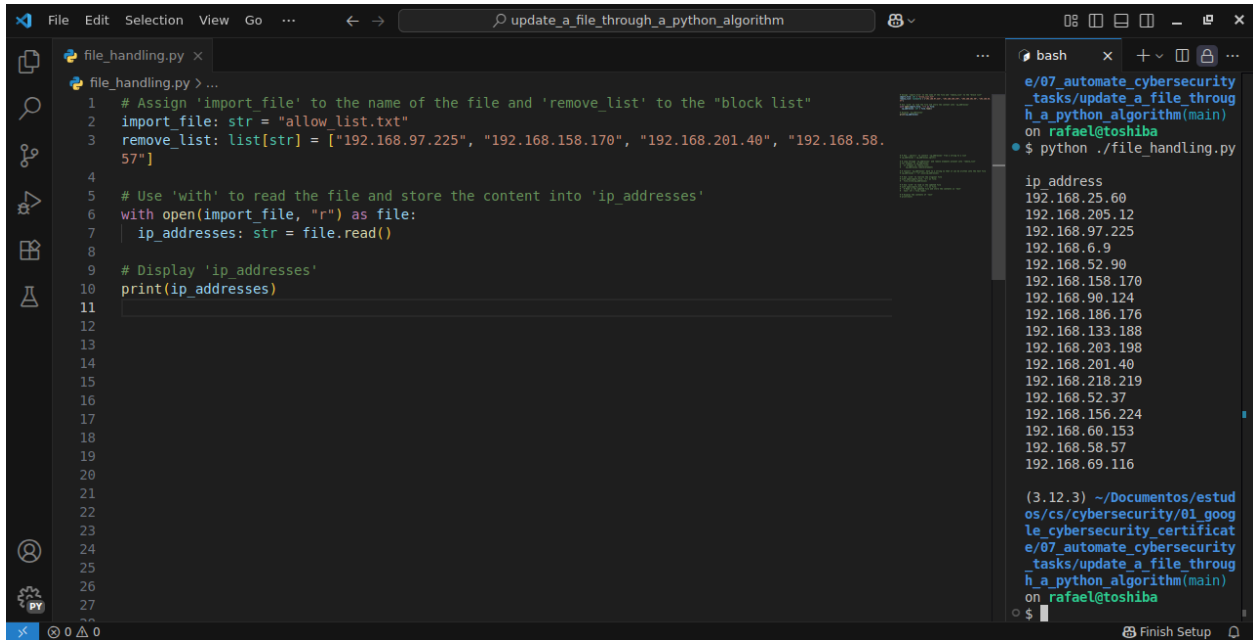
Open the file that contains the allow list



The screenshot shows a code editor with a dark theme. The file explorer on the left shows a project named 'update_a_file_through_a_python_algorithm' containing two files: 'file_handling.py' and 'allow_list.txt'. The 'allow_list.txt' file is open in the editor, displaying a list of IP addresses. The status bar at the bottom indicates the cursor is at line 19, column 1, with 4 spaces, UTF-8 encoding, LF line endings, and Plain Text format.

```
1 ip_address
2 192.168.25.60
3 192.168.205.12
4 192.168.97.225
5 192.168.6.9
6 192.168.52.90
7 192.168.158.170
8 192.168.90.124
9 192.168.186.176
10 192.168.133.188
11 192.168.203.198
12 192.168.201.40
13 192.168.218.219
14 192.168.52.37
15 192.168.156.224
16 192.168.60.153
17 192.168.58.57
18 192.168.69.116
19
```

Read the file contents



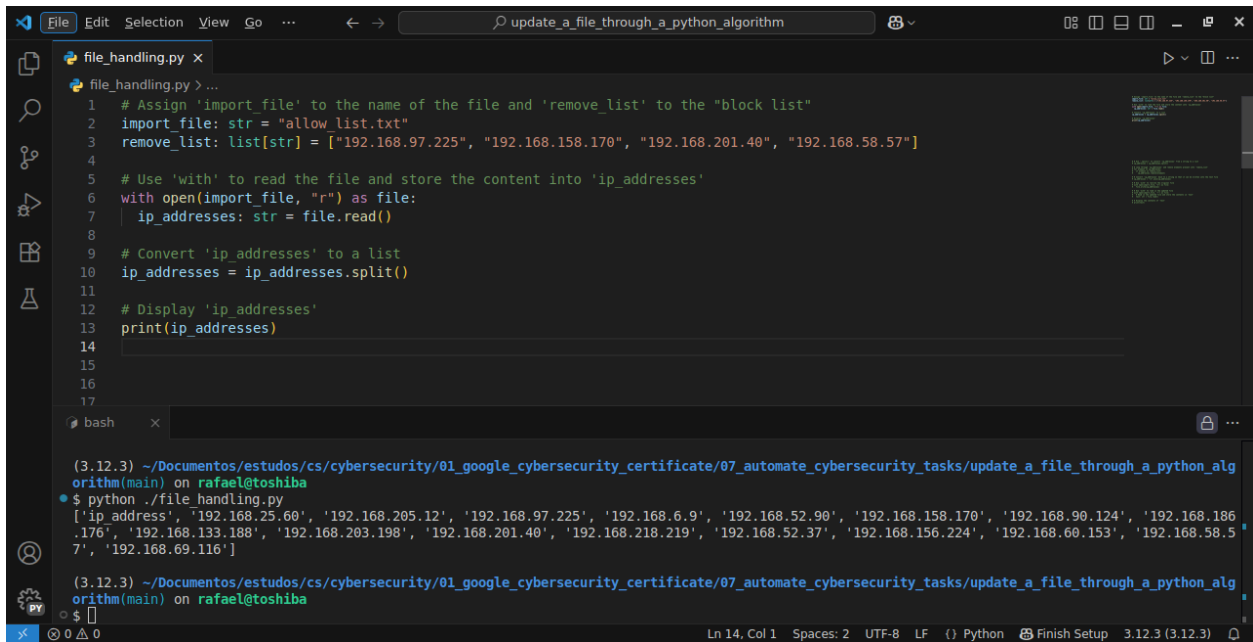
The screenshot shows a code editor with a file named `file_handling.py`. The script reads a file named `allow_list.txt` and prints its contents. The output in the terminal is a list of IP addresses.

```
1 # Assign 'import_file' to the name of the file and 'remove_list' to the "block list"
2 import_file: str = "allow_list.txt"
3 remove_list: list[str] = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
4
5 # Use 'with' to read the file and store the content into 'ip_addresses'
6 with open(import_file, "r") as file:
7     ip_addresses: str = file.read()
8
9 # Display 'ip_addresses'
10 print(ip_addresses)
```

The terminal output shows the following 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
```

Convert the string into a list



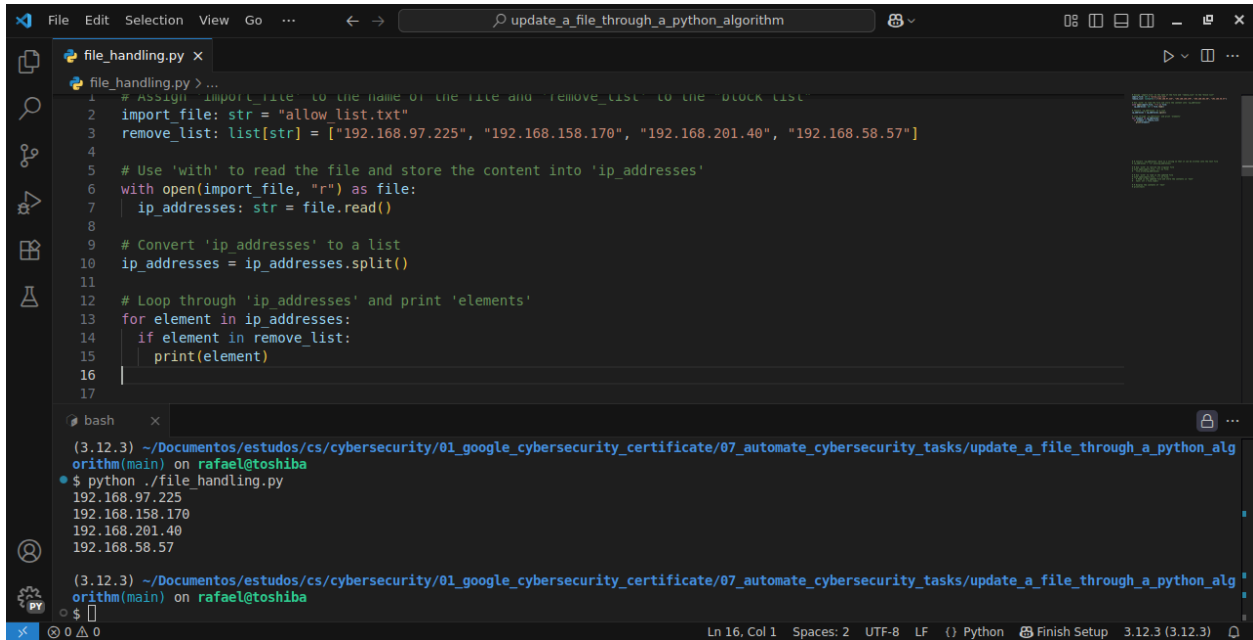
The screenshot shows the same code editor with the `file_handling.py` script. The script now converts the string `ip_addresses` into a list using the `split()` method. The output in the terminal is a list of IP addresses as strings.

```
1 # Assign 'import_file' to the name of the file and 'remove_list' to the "block list"
2 import_file: str = "allow_list.txt"
3 remove_list: list[str] = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
4
5 # Use 'with' to read the file and store the content into 'ip_addresses'
6 with open(import_file, "r") as file:
7     ip_addresses: str = file.read()
8
9 # Convert 'ip_addresses' to a list
10 ip_addresses = ip_addresses.split()
11
12 # Display 'ip_addresses'
13 print(ip_addresses)
```

The terminal output shows the following IP addresses as a list of strings:

```
['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']
```

Iterate through the remove list



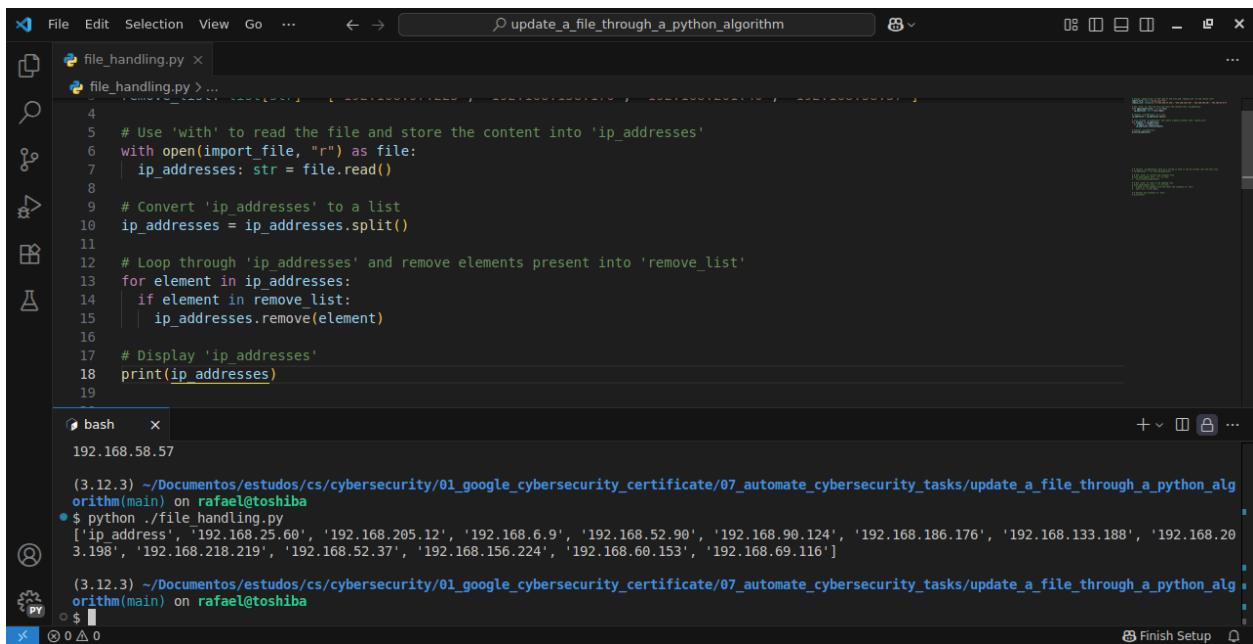
The screenshot shows a VS Code editor with a file named `file_handling.py`. The script defines a `remove_list` and reads IP addresses from `import_file.txt`. It then iterates through the `ip_addresses` list and prints any IP address that is also in the `remove_list`.

```
1 # Assign 'import_file' to the name of the file and 'remove_list' to the 'block list'
2 import_file: str = "allow_list.txt"
3 remove_list: list[str] = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
4
5 # Use 'with' to read the file and store the content into 'ip_addresses'
6 with open(import_file, "r") as file:
7     ip_addresses: str = file.read()
8
9 # Convert 'ip_addresses' to a list
10 ip_addresses = ip_addresses.split()
11
12 # Loop through 'ip_addresses' and print 'elements'
13 for element in ip_addresses:
14     if element in remove_list:
15         print(element)
16
17
```

The terminal output shows the execution of the script, which prints the IP addresses that are in the `remove_list`:

```
(3.12.3) ~/Documentos/estudos/cs/cybersecurity/01_google_cybersecurity_certificate/07_automate_cybersecurity_tasks/update_a_file_through_a_python_algorithm(main) on rafaelt@toshiba
$ python ./file_handling.py
192.168.97.225
192.168.158.170
192.168.201.40
192.168.58.57
```

Remove IP addresses that are on the remove list



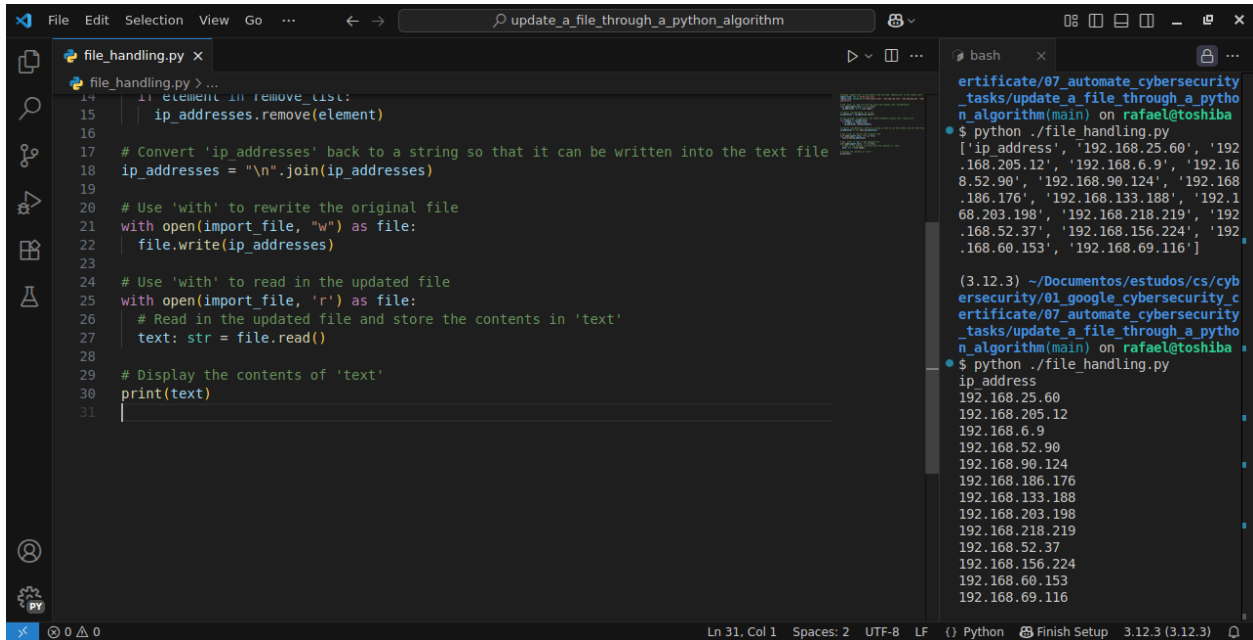
The screenshot shows a VS Code editor with a file named `file_handling.py`. The script defines a `remove_list` and reads IP addresses from `import_file.txt`. It then iterates through the `ip_addresses` list and removes any IP address that is also in the `remove_list`. Finally, it prints the remaining IP addresses.

```
4
5 # Use 'with' to read the file and store the content into 'ip_addresses'
6 with open(import_file, "r") as file:
7     ip_addresses: str = file.read()
8
9 # Convert 'ip_addresses' to a list
10 ip_addresses = ip_addresses.split()
11
12 # Loop through 'ip_addresses' and remove elements present into 'remove_list'
13 for element in ip_addresses:
14     if element in remove_list:
15         ip_addresses.remove(element)
16
17 # Display 'ip_addresses'
18 print(ip_addresses)
19
```

The terminal output shows the execution of the script, which prints the IP addresses that are not in the `remove_list`:

```
(3.12.3) ~/Documentos/estudos/cs/cybersecurity/01_google_cybersecurity_certificate/07_automate_cybersecurity_tasks/update_a_file_through_a_python_algorithm(main) on rafaelt@toshiba
$ python ./file_handling.py
192.168.58.57
['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']
```

Update the file with the revised list of IP addresses



The screenshot shows a code editor with a file named `file_handling.py` and a terminal window. The Python script in the editor performs the following steps:

- Iterates over `remove_list` and removes each element from `ip_addresses`.
- Converts the `ip_addresses` list back to a string using `"\n".join(ip_addresses)`.
- Uses `with open(import_file, "w") as file:` to rewrite the original file with the updated `ip_addresses`.
- Uses `with open(import_file, 'r') as file:` to read the updated file and store its contents in a variable `text`.
- Displays the contents of `text` using `print(text)`.

The terminal window shows the execution of the script, displaying the updated list of IP addresses:

```
ertificate/07_automate_cybersecurity_tasks/update_a_file_through_a_python_algorithm(main) on raphael@toshiba
$ python ./file_handling.py
['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']
(3.12.3) ~/Documentos/estudos/cs/cybersecurity/01_google_cybersecurity_certificate/07_automate_cybersecurity_tasks/update_a_file_through_a_python_algorithm(main) on raphael@toshiba
$ python ./file_handling.py
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
```

Summary

`update_file(import_file, remove_list)`

Provide a file name (the application needs to be executed within that file's folder) and a list of denied or blocked IPs for removal. Through file management using “with” and “open()” all operations are safely executed, upgrading the file contents.