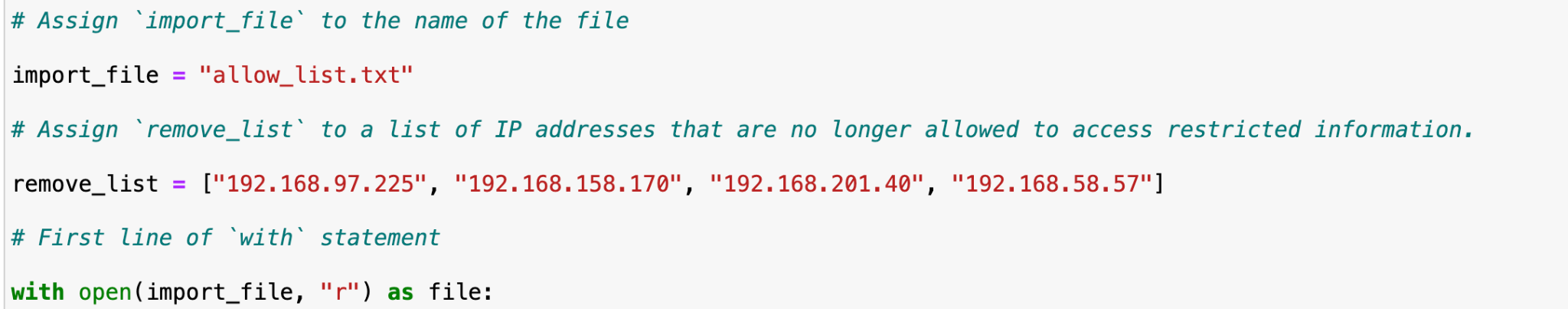
# Python algorithm for updating an allow list

## Project Description

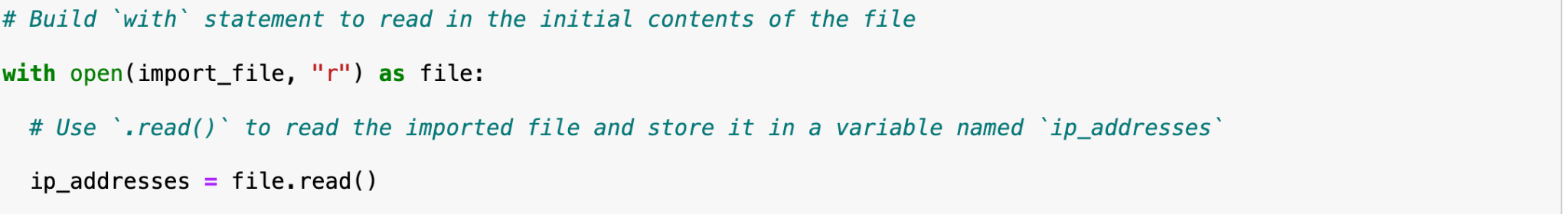
## Develop an automated algorithm using Python to keep the allow list up to date by scanning and removing unauthorized IP addresses to ensure PoLP (principle of least privilege).

## *Open the file that contains the allow list*



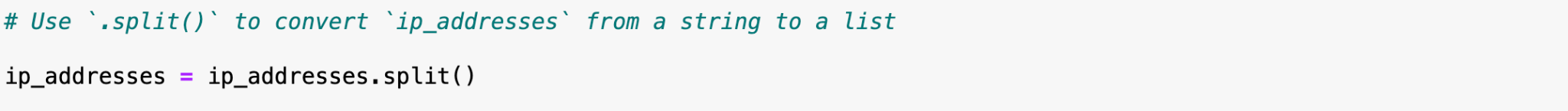
‘with’ is a statement that will self-close the function when finished. ‘open()’ begins a function that will open the selected file, while “r” signals that the function intends to make the file readable. This will be saved as “file” for the function’s purposes.

## *Read the file contents*



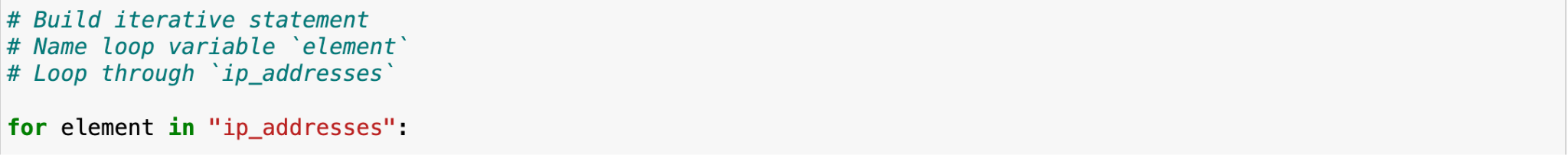
‘File’ will now be able to be read, using the method ‘.read()’. It is assigned to the variable now known as “ip\_addresses”.

## *Convert the string into a list*



Using the ‘.split()’ method, the “ip\_addresses” file is converted from a string into a list.

## *Iterate through the remove list*



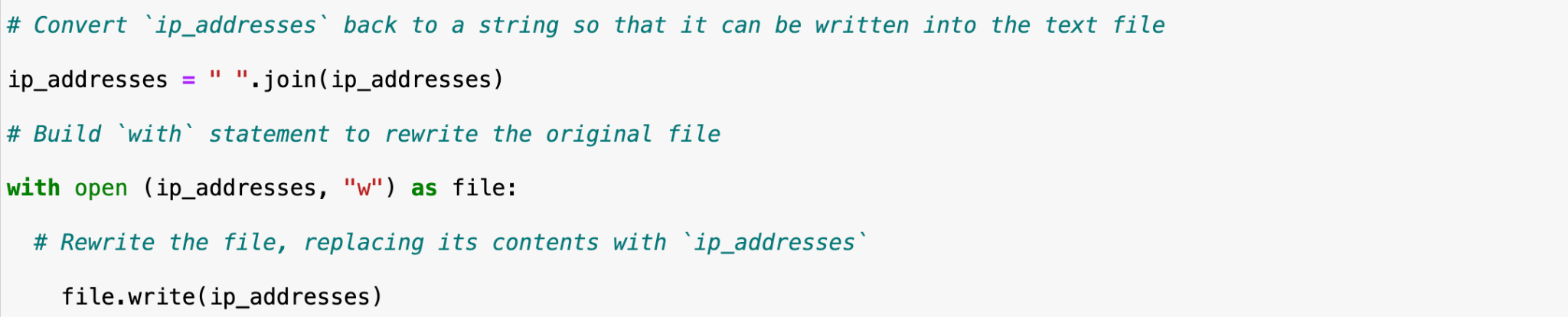
An iterative (repeating) loop is built using ‘for’. The objects in the approved “ip\_addresses” list are assigned the variable name “element”, so each element will be scanned in the loop for verification.

## *Remove IP addresses that are on the remove list*



The loop is then directed to scan the elements (in the currently approved “ip\_addresses” list) against those in the “remove\_list”. If an element is found matching one in the “remove\_list”, the algorithm is directed to remove the matching element from the approved list in “ip\_addresses” using the ‘.remove()’ method.

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



“ip\_addresses” is converted back from a list to a text file using ‘.join()’. The algorithm is then coded to update the approved “ip\_addresses” file using ‘with open’ and the ‘.write()’ method.

## 

## Summary

By identifying unauthorized IP addresses and removing them quickly and efficiently, vulnerabilities that could lead to unauthorized access, data breaches, and system disruption are averted.