



Cybersecurity (Python) Lab Journal - Entry 4

Lab Description:

This lab was about using Python to update a file that stores IP addresses for people who have access to a secure healthcare subnet. The task was to take a file called `allow_list.txt`, check it against another list called `remove_list`, and automatically delete any IPs that matched.

Date: Record the date of the journal entry.	Entry: #4 Date: November 7, 2025
Description	I wrote a small Python script that cleaned up the allow list file. The idea was to open the file, read all the IPs, and compare them with another list that has the IPs that should be removed. If any matched, they got deleted, and then the file was updated with the new list.
Tool(s) used	Python 3, text editor
Final Code	<pre>import_file = "allow_list.txt" with open(import_file, "r") as file: ip_addresses = file.read().split("\n") remove_list = ["192.168.1.10", "172.16.0.5", "10.0.0.11"] for element in remove_list: if element in ip_addresses: ip_addresses.remove(element)</pre>

	<pre>with open(import_file, "w") as file: file.write("\n".join(ip_addresses))</pre>
Test / Result:	<ul style="list-style-type: none"> Before file had: 192.168.1.10, 172.16.0.5, 10.0.0.11, 10.0.0.20, 172.16.0.8 After running: 10.0.0.20, 172.16.0.8 (removed the three matches)
Key Findings:	<p>Key Findings / Skills Practiced:</p> <ul style="list-style-type: none"> File I/O with <code>with open(..., "r"/"w")</code> Converting text ↔ list using <code>.split("\n")</code> and <code>"\n".join()</code> Basic list ops (<code>in</code>, <code>.remove()</code>) to handle allow/deny maintenance Keeping formatting (one IP per line) when writing back
Additional Notes	Assumes no duplicate IPs (so <code>.remove()</code> is fine). For production I'd back up the file and log removed IPs for audit.
