Scriptspråk

Kurs_NÄTD24LIN_SCSP25

Workshop 1: Network Management System (NMS) - Data Export

1. Problem Solving and Challenges :-

What were the biggest challenges you encountered in this workshop part?

- This is my first time writing Python code and I found it challenging to read data from JSON files initially. The data contained various types of devices, each with different properties like ports, status, and VLANs.

- As a newcomer to Python libraries, I struggled to understand where and how to use various functions. However, after conducting online research, I learned about utilizing libraries and functions effectively.

How did you solve them?

- I used structured loops and condition checks to sort and count each type of device (switches, routers, etc.).
- I created separate counters and summaries to handle multiple statistics at once.
- For example, I used this block to calculate devices with low uptime:

```
if(device["uptime_days"] < 30):
    low_up_time_devices += 1
    critical = " ⚠ critical" if device["uptime_days"] <= 2 else location["site"]
    low_up_time_info += device["hostname"] + "      " + str(device["uptime_days"]) + "days  " + critical + "\n"
```

This helped identify unstable devices effectively.

2. Learning and Development: -

What have you learned that you couldn't do before?

- I learned how to parse JSON data in Python and use it for real-world network reports.
- I also learned how to calculate percentages, format outputs, and handle conditional alerts (like low uptime or high port usage).

Which concept(s) was/are the most difficult to understand and why?

- Understanding how to handle nested JSON structures was tricky at first, especially when accessing device attributes like ["ports"]["used"] or optional fields like vlans.

- Also, managing all counters and conditions without mixing them up required careful logic and testing.
- To debug and print values, I used print statements to display results on the screen. I also discovered the benefits of formatting output in tables using f-strings.

3. Professional Relevance: -

How can you use these skills in your future role as a network engineer?

- As a network engineer automation like this saves time and reduces human error.
- Instead of checking each device manually this script can give a full picture of the network status in seconds.

Examples of real-world situations where this type of automation would be valuable:

- If a switch goes offline, it's flagged automatically in the report.
- If port usage exceeds 80%, the code adds a ⚠️ warning so you can plan upgrades.
- Access points with too many clients (≥45) are highlighted as overloaded, helping with capacity planning.
- Keeping track of device uptime to schedule maintenance or replacements.

4. Code Quality and Improvements: -

If you were to do the workshop part again, what would you do differently?

- My first script was straightforward, but I recognize the importance of improving code quality.
- Break down the script into reusable functions to enhance maintainability and readability.
- Eliminate redundant code and leverage functions to streamline the script.
- I'd also add error handling for missing data (like missing vlans or ports) to make it more robust.

Which parts of your solution are you most satisfied with and why?

- I'm most satisfied with the executive summary and report formatting section. It combines technical data into a readable, human-friendly format, including percentages, alerts, and clear section headers. It turns raw network data into an actionable summary that anyone in IT could understand without digging into JSON files.