Student Name: **Weight:** **20%**

Student ID: **Marks:** **/24**

Assignment: Designing a Network

Working individually, read the scenario below and answer the questions that follow. Your overall goal is to design a network that addresses the needs of the client in the scenario.

Scenario

You are working for client who needs to relay important information from client machines and a server machine to a central location where this information can be processed and monitored. Performing these tasks manually would take a lot of time, so automation is the best approach in this situation. The Client machine has regularly occurring cron jobs that run two python scripts in sequence. The cron jobs should run at boot (the SysAdminTask.py runs followed by Client.py). The company wants you to recommend the best technologies for this purpose.

Instructions

1. You will use 2 virtual machines for this project.
    - Machine 1 = Admin Machine (OS can be Linux or Windows)
    - Machine 2 = Client Machine (OS must be Linux)
2. Proper error handling should be in place to ensure that the SysAdminTask.py has been run and will produce the correct output file before running the Client.py script.
3. You can choose what the SysAdminTask.py script does, as long as you justify your choice. This will be part of your grade.
4. Consider appropriate system administrator tasks: What information would an administrator want to collect?
5. When the Client Machine is rebooted, it will automatically send a file over the network to the admin machine which has a python server script waiting to receive files. The file will contain data collected from resident script on the Client machine.

Deliverables

- Your network should reflect a good design approach and demonstrate fundamental network configuration standards.
- You must use 3 scripts:
    - Server.py
    - Client.py

- SysAdminTask.py

- Submit a 10-minute demo explaining your code and demonstrating your success.
- Submit a document in which you answer the following questions regarding your application of SysAdminTask.py:
  - What is the purpose of the script?
  - What business need does it meet?

Important: Your work will be reviewed for similarities within and across cohorts.

Tips for Success

- Create the SysAdminTask.py script that will produce a file with information that would be useful to a system administrator. The script will receive the content on the Server side of the network.
- The content should be concise and organized, so the SysAdminTask.py script should only contain useful information. For example, you don't want the script to send all of the logs and system information.
- Make sure the Client Machine can send the output for the SysAdminTask.py script to the Server Machine manually. Once this is confirmed, set up the crontab on the client machine to schedule the SysAdminTask.py, followed by the Client.py scripts to run at reboot.
- Proper testing is the key to success for this task. For this reason, you need to ensure that the content exists. You can do this by deleting the output file upon testing, and then opening the output file on the server machine.

Marking Criteria

Categories	0	1	2	Score
SysAdminTask.py	Not functioning	Runs without errors but does not run automatically	Runs correctly as a cron job	**/2**
SysAdminTask.py Code Content	Not functioning	Minimal code with limited useful functionality for system administrators	Well-defined code	**/2**

SysAdminTask.py Functionality	Not functioning	Minimal sys admin content in output file. Output content is not helpful for system admins when performing administrative tasks on the particular system.	Usable sys admin content in output file	**/2**
SysAdminTask.py Justification	Not functioning	Minimal justification (justification does not explain why the script is required)	Well-explained: What is the purpose of the script? What business need does it meet?	**/2**
Client.py can connect to server	Not functioning	Attempted (code exists but is not fully implemented and may not be working in all cases)	Runs correctly, as proven in demo	**/2**
Client.py Sends File to server	Not functioning	Attempted (code exists but is not fully implemented and may not be working in all cases)	Runs correctly, as proven in demo	**/2**
Server.py accepts connections	Not functioning	Attempted (code exists but is not fully implemented and may not be working in all cases)	Runs correctly, as proven in demo	**/2**
Server.py Receives File	Not functioning	Attempted (code exists to accept the file but the file is not created properly)	Runs correctly, as proven in demo	**/2**
Cron job runs SysAdminTask.py	Not functioning	Attempted with demo showing that the job was created but does not execute as expected	Runs correctly at reboot, as proven in demo	**/2**

Cron job runs Client.py	Not functioning	Attempted with demo showing that the job was created but does not execute as expected	Runs correctly at reboot, as proven in demo	**/2**
Error Handling	None	Limited (missing error handling techniques when working with resources, casting data types, etc.)	Well-executed	**/2**
Code Readability	Insufficient (very difficult to follow the coding solution, lacks organization and logical structure)	Missing concise pseudocode, missing sufficient code commenting, or missing proper code structure	Code is easy to understand. Provides proper documentation.	**/2**
			Total	**/24**