Name:

Course:

Affiliation:

Date Submitted:

**Part 1**

**I.a) Detailed Installation and Usage Guide**

The set up and utilization of the "input.sh" script for monitoring your system.

1. Make a script save:

* Launch a textual content editor (including vi, nano, or your favorite code editor that still has text enhancing capabilities).
* The script content material you received (or modified) ought to be copied or written into the editor.
* Save the document as "input.sh" and put it in a convenient location.

2. Turn on the script's execution:

* Get a terminal window open.
* To find the directory where you stored the script, use the cd command. For instance, enter cd if you saved it in your house directory.
* To give the script execution rights, use the chmod command. The script can be accomplished via the owner consumer with the subsequent command:

**chmod +x input.sh**

* Make certain the character executing the script has the potential to write down to the listing containing output.html. chmod may be used to view and alter permissions:

**chmod +w /path/to/output.html**

* You can increase your permissions with the aid of the use of the chmod command with sudo when you have sudo privileges:

**sudo chmod +w /path/to/output.html**

3. Execute the Script manually:

* If you're now not already there, navigate to the script's listing inside the terminal window.
* To run the script, input the following command:

**./input.sh or sudo ./input.sh**

* The "output.log" file inside the equal listing will incorporate statistics logged by using the script as it runs and monitors the system.
* Script may be running inside the Background: If there may be no output displayed in the terminal, the script may be operating inside the history. To verify whether facts is being created and up to date inside the output.log file. To hold an eye at the very last few lines of the log document, use the tail command:

**tail -f output.log**

This command will continuously display the last few lines of the log file as they are being written.

* To locate the ‘output.log’ file, use ‘ls’ command to list the files in the current directory.

**ls -l output.log**

4. Cron-assisted automated execution (optional):

With Linux, you could schedule jobs to execute automatically at predetermined durations or durations the usage of the Cron application.

You need to change the crontab record to installation the script's automated execution via cron. Cron jobs are saved on this record in your user.

**1.b) Design Consideration**

This new script makes use of pre-installed tools, integrates modular functionalities, and places an emphasis on efficiency. Now let's examine the design decisions:

Commands used:

* free: Retrieves free memory information, preserving compatibility with the preceding technique.
* Crontab: Checks for scheduled duties, regular with the earlier script.
* Uptime: Provides device uptime and load statistics, like the original script.
* lshw: Lists precise hardware records. Chosen for its ability to offer comprehensive hardware info.
* dmesg: Displays kernel ring buffer messages, providing a view of latest device activities and mistakes.
* date: Generates timestamps for log entries, important for tracking monitoring instances.
* echo: Writes log messages to the terminal.
* awk (not explicitly used): Its functionality can be incorporated for extra granular information extraction if wished (optional future enhancement).
* grep: Search for a pattern in a file.
* sudo: Run a command with administrative privileges.
* Sudo apt install inotify-tools: The inotifywait command may not be available to your machine, as proven with the aid of the error inotifywait: command now not found. The inotify-tool package deal, of which inotifywait is part, might not be installed by default on some structures.

Writing the Script:

* Modular Functions: The script has been divided into well-defined functions.
* Log\_info: Manages the timestamp-primarily based logging of statistics to the output file.
* Monitor\_tasks, monitor\_memory: By focusing on a single tracking characteristic, every function enhances the readability and company of the code.
* main: The fundamental feature presents a centralized manage factor via continuously executing all monitoring capabilities.
* Error Handling: If there are not any cron jobs, the monitor\_tasks function looks for them and units a default message. Thus, blank log entries are prevented.
* Looping and Scheduling: To constantly run tracking workouts at ordinary intervals (controlled by means of the sleep command), the main function uses some time proper loop. This lets in continuous device remark.

**Justification**

* Pre-installed Tools: Like the preceding script, this one is also based on extensively available instructions, this means that that it's going to paintings with a more variety of Linux distributions.
* Modular Design: The script encourages code reuse, maintainability, and clean comprehension with the aid of segmenting functionalities into features.
* Error Handling: To save you sudden behavior when there are no cron jobs, the script suggests primary errors handling.
* Flexibility: New tracking features can be brought inside the destiny even as adhering to the modern-day pattern, thanks to the modular creation.

**Why I chose these techniques.**

The script makes use of pre-installed functions such as free and uptime for wider compatibility, setting simplicity and efficiency first. Code clarity and maintainability are advanced by means of modular capabilities for every monitored thing (memory, responsibilities, hardware, logs, load). The script avoids superfluous clutter within the log report while taking pictures important records. New tracking functions can be without problems introduced to this modular layout, and fundamental blunders managing ensures beneficial output. It is essential to apply sudo to run with root privileges to get entry to machine sources. This method offers flexibility to alter the intervals of monitoring and the extent of log detail consistent with your necessities, so presenting a sturdy foundation for fundamental device tracking.

**1.c) Extensive Test results and Sample output**

Test Environment:

Operating System: Ubuntu 22.04LTS.

Script: The newly created script in vscode, with modular functions and monitoring capabilities.

Test Procedure:

1. Saved the script as “input.sh”.

2. Made the script executable using chmod + x input.sh.

3. Run the script with root privileges: sudo ./input.sh.

Expected Output:

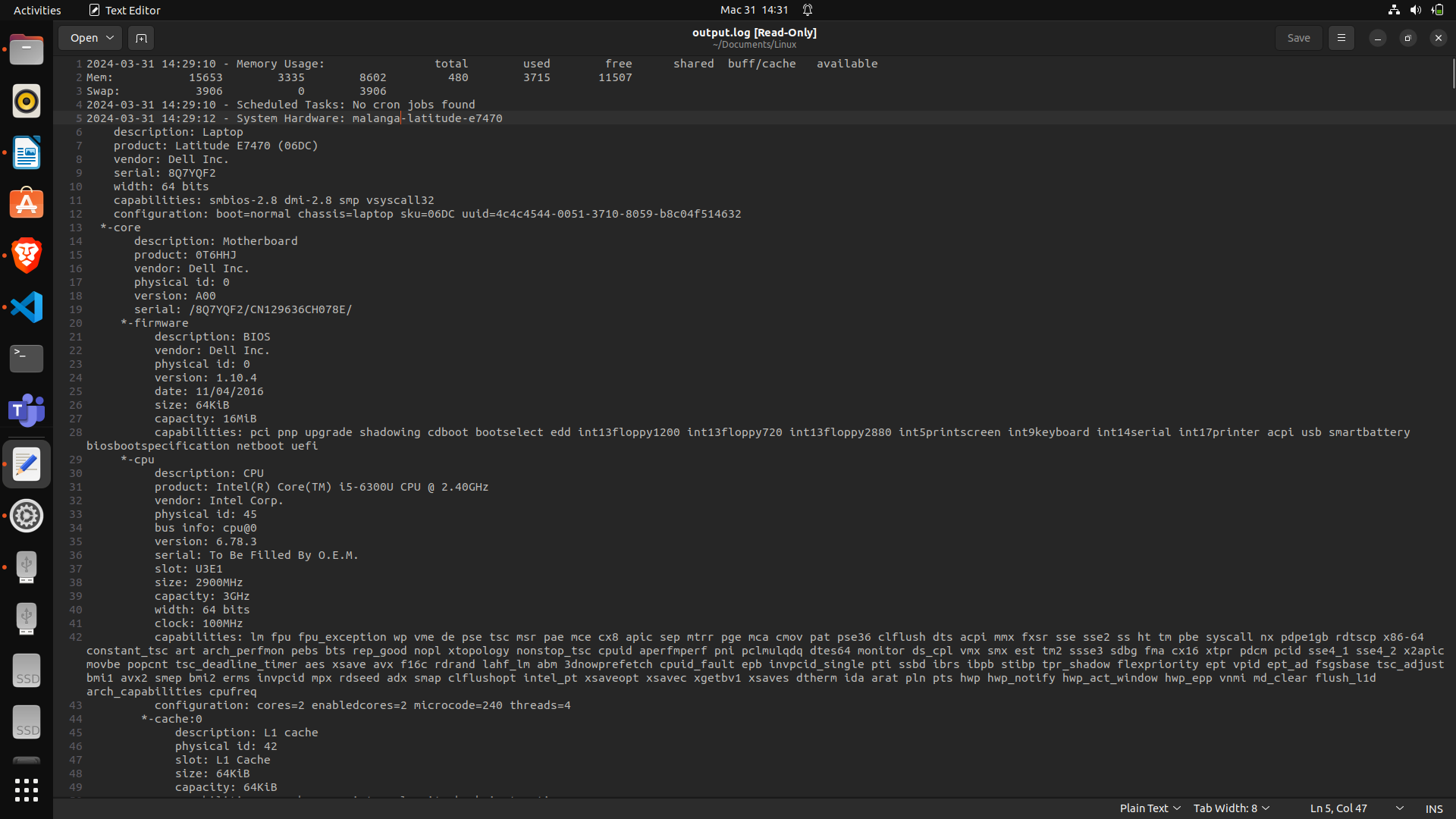
Free Memory: The available amount of memory in MB.

Scheduled Tasks: List of present cron jobs, or a message indicating no cron jobs found.

System Hardware: A summary of hardware information retrieved using lshw.

System logs: Recent kernel ring buffer messages captured by dmesg.

System Load: Current system load information displayed by uptime.



The verbosity of the logs is controlled by way of a variable referred to as LOG\_LEVEL inside the script. Three log degrees are furnished via the script:

* Standard: Basic records that is typically pertinent for monitoring reasons is logged at this degree.
* Verbose: This stage logs greater unique records, which includes more history or motives.
* Debug: This degree logs even more facts in-intensity, regularly includes debugging info which might be useful for troubleshooting but may not be required for routine tracking.

The LOG\_LEVEL variable is used as follows:

When LOG\_LEVEL is set to verbose or standard, the script reviews messages in HTML format along with timestamps.

The script logs timestamps and HTML-formatted messages whilst LOG\_LEVEL is ready to debug; however, the messages are shown in gray.

The LOG\_LEVEL variable can be modified to fit your needs and preferences. More thorough logs could be produced, for instance, if you set it to verbose; even greater specified logs could be produced if you set it to debug. Just normal monitoring records can be logged in case you set it to standard.

**1.d) Conclusion and reflection- Advantages and Disadvantages**

The supplied script affords a useful approach for essential device monitoring that has the following benefits:

**Benefits**

* Efficiency and Simplicity: The script makes use of already set up equipment, so it doesn't require any greater dependencies and works with one of a kind Linux distribution. Maintainability and readability are advanced by modular capabilities.
* Concentrate on Important Information: The script statistics pertinent data for each monitored element, ensuring that the log file is informative and free of superfluous information.
* Flexibility and Customization: Adding new tracking functions is made easy with the aid of the modular design. Meaningful output is ensured by means of basic error control. The log degree and monitoring c language can be changed to fulfill your desires.
* Lightweight and Non-Intrusive: The script minimizes its impact at the machine with the aid of the use of without difficulty available assets and avoiding the installation of additional programs.

**Drawbacks**

* Restricted Scope: The script concentrates on memory, tasks, hardware, logs, and cargo tracking. Additional equipment or scripts may be required for complete tracking of sources (which includes CPU temperature and disk I/O).
* Security Implications: For positive tracking activities, the script ought to be completed with root privileges. To keep away from any safety risks associated with extended privileges, it is imperative to apply caution and most effective run the script when vital.
* Restricted Log Analysis The script records unprocessed statistics. Additional scripting or specialized log analysis tools may be needed for superior log analysis or for parsing complicated log formats.

**Overall Reflection**

This script gives a beneficial starting point for essential system monitoring. Its flexibility, performance, and simplicity of use make it an easy-to-use tool for studying about the fitness of your device. Even though it may not offer huge monitoring abilities, it's a very good location to begin. To accommodate greater specialized monitoring necessities, the script can be accelerated and altered.

You can use this script as a constructing block to expand a greater entire and customized system tracking answer by contemplating its advantages and drawbacks.

**References**

*free(3): allocate/free dynamic memory - Linux man page*. (n.d.). Linux.die.net. <https://linux.die.net/man/3/free>

*GNU Grep 3.7*. (n.d.). Www.gnu.org. <https://www.gnu.org/software/grep/manual/grep.html>

Middleton, J. (n.d.). *logcheck(8) - Linux man page*. Linux.die.net. <https://linux.die.net/man/8/logcheck>

Okoi, M. D. (2019, May 31). *Linux Uptime Command With Usage Examples*. Www.tecmint.com. <https://www.tecmint.com/linux-uptime-command-examples/>

Ramey, C. (2022). *Bash Reference Manual*. <https://www.gnu.org/software/bash/manual/bash.pdf>

Sharma, S. (2023, July 14). *lshw Command in Linux: Get Hardware Details*. Linuxhandbook.com. <https://linuxhandbook.com/lshw-command/>

*The GNU Awk User’s Guide*. (n.d.). Www.gnu.org. <https://www.gnu.org/software/gawk/manual/gawk.html>

Vixie, P., Mašláňová , M., Dean, C., & Mráz, T. (2013, September 26). *cron(8) - Linux manual page*. Man7.org. <https://man7.org/linux/man-pages/man8/cron.8.html>

**Part 2**

**Question 1**

Safiol Pharmaceuticals' Secured Network Solution

This design addresses top call for intervals, failover mechanisms, faraway get right of entry to, redundancy, backups, and performance so that you can offer a steady network answer for Safiol Pharmaceuticals' UK workplace and their studies middle in Japan.

Components of the Network:

UK Office

Core Switch: High-performance switch for managing traffic within the critical network.

Distribution switches are people who hyperlink the main switch to departments together with administration and studies and development.

Access switches are the switches that link positive devices, consisting of PCs and servers, to the department's community.

Two firewalls configured in an excessive-availability (HA) setup are used for intrusion prevention and internet access control.

Web server: A secure and stable net server for institution research.

Email Server: A haven for correspondence through e-mail.

Data Servers (2): For excessive availability and disaster healing, redundant facts servers with reflected facts are used.

User PCs: R&D personnel workstations.

Japan Office:

High-bandwidth net connection is to be had.

Security Appliance: Multipurpose tool that mixes VPN gateway for secure far flung get entry to, firewall, and intrusion detection/prevention gadget (IDS/IPS).

User PCs: Japanese researchers' workstations.

Connection between offices:

VPN: A stable virtual private network tunnel that allows offices within the UK and Japan to talk securely.

A rationalization of network layout

1. The R&D network phase might be physically isolated from the relaxation of the organization’s network through using Virtual Local Area Networks (VLANs) to enhance security.

2. Three security zones can be hooked up within the network:

* Demilitarized Zone (DMZ): The e-mail and web servers might be placed in the DMZ, with confined get right of entry to the inner community however complete net get entry to.
* Internal Network: User PCs, R&D information servers, and internal resources will all be housed on this location.
* Japan Office Network: A VPN might be used to hyperlink this area to the United Kingdom workplace, to be able to be blanketed through the safety appliance.

3. Firewalls: By filtering and stopping malicious conduct, the twin firewalls inside the HA configuration will provide a single point of manipulate for both incoming and outgoing visitors.

4. Data Servers: In the occasion of a server failure, information redundancy will be ensured through configuring the 2 data servers for statistics mirroring. For disaster restoration functions, it's miles vital to have ordinary backups in region.

5. Remote Access: To create a steady tunnel and gain access to permitted sources on the UK R&D community, Japanese researchers will use a VPN software. Encrypted verbal exchange and user authentication are assured by using this.

6. Scalability: To handle intervals of peak demand, the community design may be easily expanded horizontally through including greater switches and servers as needed.

7. Performance: During height hours, seamless community visitors float may be assured by way of high-overall performance center and distribution switches. Furthermore, the Japan office will advantage from dedicated internet potential if you want to enhance remote verbal exchange.

8. Mechanisms for failover:

* Hardware failover: In the occasion that a primary device fails, redundant records servers and excessive-availability firewalls will right away transition to a secondary device.
* Software Failover: To similarly boom fault tolerance, load balancing or clustering for crucial servers can be implemented.
* Cloud Failover: In the event of an extensive outage at the UK office, think about a hybrid technique in which crucial information is replicated to a safe cloud garage answer for disaster healing.

Advantages:

Security: To improve universal community protection, community segmentation, firewalls, and steady remote get right of entry to techniques are used.

Redundancy: To guarantee device availability, data mirroring and failover strategies are used.

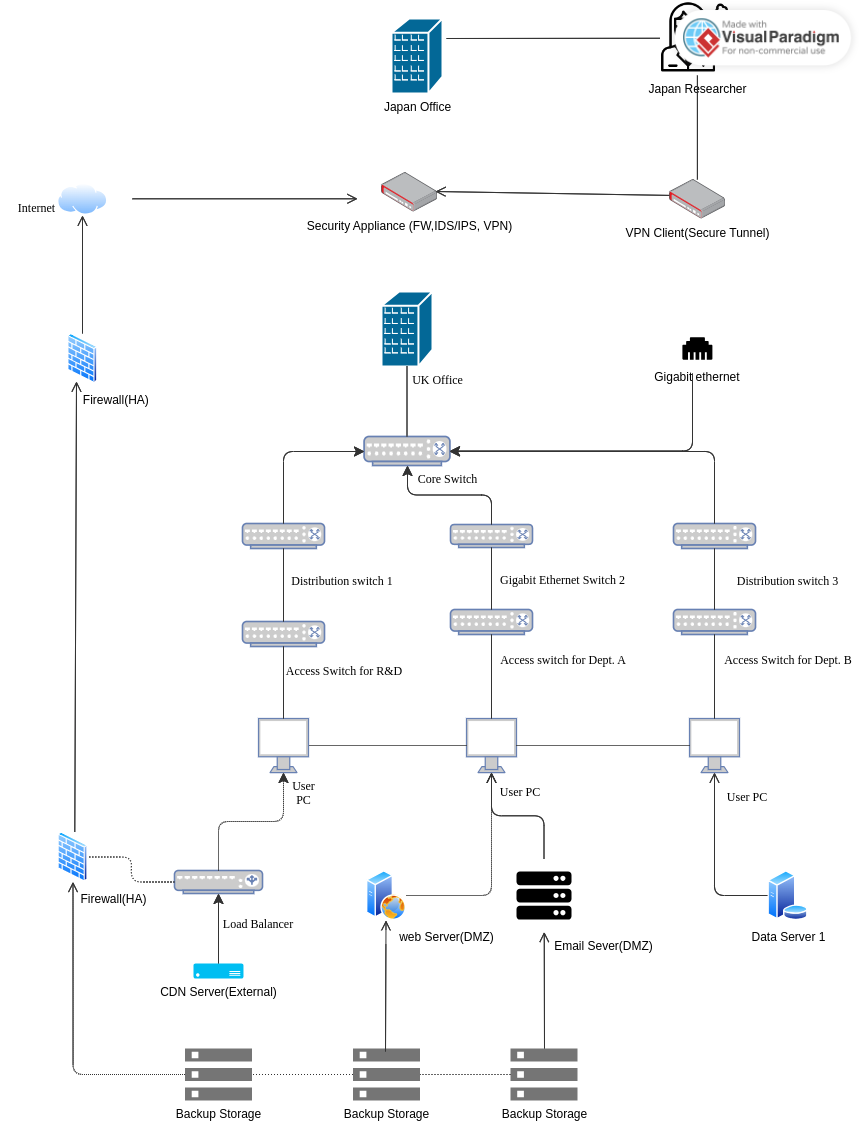
Scalability: The structure can develop and meet top call for.

Performance: Dedicated bandwidth and excessive-overall performance hardware enhance community overall performance.

Lower Administration Cost: Remote workplace control is made less difficult inside the Japan office through utilizing a multi-characteristic security equipment.

Diagram of the Safiol Pharmaceuticals Network

The secure community solution for the UK and Japan places of work of Safiol Pharmaceuticals is shown on this diagram:



**Explanation**

A safety appliance that serves as a firewall, intrusion detection/prevention device (IDS/IPS), and VPN gateway is used to hyperlink the Japan office to the net.

To create a steady tunnel and advantage get right of entry to accredited assets inside the UK R&D community, Japanese researchers use a VPN software.

An excessive-performance core switch related to distribution switches powers the office community inside the United Kingdom. Access switches in each branch (R&D, Administration, and so on.) are connected to those distribution switches, and these switches are related to equipment (PCs, servers).

Both incoming and outgoing traffic is managed via high-availability firewalls.

The Demilitarized Zone (DMZ) homes the e-mail and net servers, making an allowance for restricted net access.

To provide records redundancy, statistics servers are set up for statistics mirroring. For catastrophe healing, backups can be saved in a secure cloud garage account or regionally.

Gigabit Ethernet switches: The term "Gigabit Ethernet" is used to identify the middle and distribution switches, emphasizing their ability to manage huge quantities of site visitors.

Load Balancer: Next to the firewall, an icon of a load balancer is added. This indicates the distribution of visitors at some point of peak hours among several net servers (not expressly displayed).

Content Delivery Network (CDN): An external CDN server is proven as being online. This is a illustration of putting off static internet site content from Safiol's internet server, which enhances overall performance in the course of durations of high call for.

**Question 2**

**Benefits of Two Firewalls**

* Redundancy, now and again referred to as high availability or HA, is a vital gain. Network security is maintained even if one firewall malfunctions or has hardware failure; the alternative firewall takes over without any troubles. For Safiol specially, that is vital because research facts must continually be included.
* Granular Control: Safiol can install a layered security approach with the use of two firewalls. Stricter rules may be installed one firewall for the Demilitarized Zone (DMZ) to modify the research network's access to the Web and electronic mail servers. The R&D department's internal network traffic can be controlled via the opposite firewall, which may additionally allow extra lax guidelines for statistics sharing between researchers.
* Firewalls may be installed to load balance or split up incoming traffic among themselves. This is specifically useful at busy instances while a whole lot of researchers can be logging into the network simultaneously. (Kaplanskiy, 2019) Firewalls save you bottlenecks and assure clean network overall performance with the aid of dispersing the burden.

**Drawbacks**

* Increased Processing Overhead: Network device like switches and firewalls need to have greater processing potential to encrypt and decrypt records packets on the community or delivery layer. This may additionally motive the overall performance of the community to lag, especially for the duration of peak periods or for large records transfers.
* Complexity of Management: Various devices at the community require the management of encryption keys and certificate for network and transport layer encryption. The network security team now faces an extra administrative burden as a result. (Kumar, 2021) Extra steps and techniques are needed to distribute and control these keys securely.
* Compatibility problems: Not every tool on a community can be able to make use of the most current encryption strategies. When trying to enforce community-wide encryption, older devices or legacy equipment might not be compatible with encryption methods, which could result in connectivity problems.

Alternative for Safiol:

Safiol can make use of data-at-relaxation encryption in mild of the constraints of network/delivery layer encryption. Sensitive records are encrypted without delay on the servers (Data Servers 1 & 2) the usage of this manner. This approach has diverse advantages:

* Decreased Processing Overhead: Since encryption and decryption take place at the servers, community overall performance isn't always as negatively impacted.
* Simplified Management: To reduce complexity for network administrators, encryption keys should be managed on the servers.
* Data Security: Encrypted data is secure even within the event that an intruder breaks into the network and access data servers.

Combination Method: For network protection and data protection, Safiol can combine firewalls with data-at-rest encryption on the servers. This gives a robust security posture without growing management burden or degrading community pace.

**References**

*Case Study: Network Modeling For Pharma*. (n.d.). Establish Inc. Retrieved March 31, 2024, from <https://www.establishinc.com/distribution-network-design-for-pharma>

Kaplanskiy, A. (2019, December 9). *Benefits of Firewall for Business*. TrustNet. <https://trustnetinc.com/firewall-for-business/>

Kumar, S. (2021, October 6). *Advantages and Disadvantages of Firewall in Computer Network*. The Crazy Programmer. https://www.thecrazyprogrammer.com/2021/10/advantages-and-disadvantages-of-firewall-in-computer-network.html