**System Health Check Script**

**Introduction**

The goal of this project was to develop a shell script capable of performing a comprehensive system health check. The script gathers information regarding disk usage, memory usage, running processes, and network connections, generating a daily report that is emailed to the system administrator. This automated process ensures continuous monitoring and quick identification of potential issues, enhancing system reliability and performance.

**Purpose**  
 The purpose of this project was to develop an automated script that would perform regular health checks on a server system and report on the status. This provides visibility into key metrics and helps proactively monitor performance and potential issues.

**Script Overview**

The shell script, named system\_health\_check.sh, performs the following key tasks:

1. **Disk Usage**: Uses df -h to display disk usage in a human-readable format.
2. **Memory Usage**: Uses free -h to display memory usage in a human-readable format.
3. **Running Processes**: Uses ps aux --sort=-%mem | head -n 10 to list the top 10 memory-consuming processes.
4. **Network Connections**: Uses netstat -tuln to display network connections.
5. **Email Report**: Generates a report and emails it to the system administrator.
6. **Cleanup**: Removes the temporary report file after sending the email.

**Implementation Steps**

1. **Save the Script**:
   * The script should be saved as system\_health\_check.sh in /usr/local/bin/.
2. **Make the Script Executable**:
3. **Schedule the Script**:
   * Use cron to schedule the script to run daily. Edit the crontab file by running:

The outputs are concatenated into a report along with descriptive headings. The mail command emails this report to the administrator while preserving any newline characters in the output.

Testing

The script was tested on an kali server. It successfully gathered all required system data and correctly formatted an email report. A daily cron job was set up to automatically run the check at midnight and email reports.

Results

Running the script provided a snapshot of key server health indicators. Any issues like high disk usage, memory pressures or unusual processes/connections could now be detected on a continuous basis. The daily reporting allows prompt monitoring and remediation if needed.

**Explanation of Script Functions**

* **Admin Email**: Specifies the email address of the system administrator to receive the report.
* **Temporary File**: Creates a temporary file to store the report contents.
* **Date and Time**: Captures the current date and time for the report header.
* **Disk Usage**: Collects disk usage information in a readable format using df -h.
* **Memory Usage**: Gathers memory usage details using free -h.
* **Running Processes**: Lists the top 10 memory-consuming processes using ps aux --sort=-%mem | head -n 10.
* **Network Connections**: Displays current network connections using netstat -tuln.
* **Email the Report**: Uses the mail command to send the report to the system administrator.
* **Cleanup**: Removes the temporary file after the report has been emailed.

**Conclusion**

The developed shell script effectively automates the system health check process, providing daily reports on critical system parameters. By scheduling this script to run daily via cron, system administrators can ensure continuous monitoring and timely intervention, maintaining optimal system performance and reliability.