sysreport.sh — A Comprehensive Linux System Diagnostic Tool

Purpose: Automatically gather system stats, health checks, disk usage, memory/cpu diagnostics, user sessions, service states, and security checks, and output a clean JSON report. Ideal for automation pipelines, cron jobs, or devops toolchains.

Features

- Modular function definitions
- Hardware + OS profiling
- Filesystem + disk usage checks
- Root access + permission checks
- Load, memory, and CPU analysis
- User login sessions
- Failed services (systemd)
- Error trapping and fallback logic
- JSON-formatted output
- Timestamped log output
- Cron-friendly and quiet mode

Full Script: sysreport.sh

bash

CopyEdit

#!/usr/bin/env bash

```
# ──────────────────────────────────────────────────────────────
# sysreport.sh - System Diagnostics Generator
# By: Your Name Here (Pro Bash Script)
# Version: 1.0.0
# License: MIT
# ──────────────────────────────────────────────────────────────

set -euo pipefail
IFS=$'\n\t'

# ──── Configuration ──────────────────────────────────────
LOG_DIR="/var/log/sysreport"
OUT_DIR="/tmp"
REPORT_NAME="sysreport-$(date +%Y%m%d_%H%M%S).json"
QUIET=false
LOG_FILE="$LOG_DIR/run-$(date +%s).log"

# ──── Colors ─────────────────────────────────────────────
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
NC='\033[0m' # No Color

# ──── Helpers ────────────────────────────────────────────
log() {
```

```
    [[ "$QUIET" = false ]] && echo -e "${YELLOW}[*] $1${NC}" | tee -a
"$LOG_FILE"
}

success() {
    echo -e "${GREEN}[+] $1${NC}" | tee -a "$LOG_FILE"
}

error() {
    echo -e "${RED}[!] $1${NC}" | tee -a "$LOG_FILE" >&2
}

# ──────── Trap Errors ──────────────────────────────────────
trap 'error "Script failed at line $LINENO. See $LOG_FILE for details."' ERR

# ──────── Initialize ───────────────────────────────────────
init_dirs() {
    mkdir -p "$LOG_DIR" "$OUT_DIR"
    touch "$LOG_FILE"
    log "Initialized directories."
}

check_root() {
    if [[ "$EUID" -ne 0 ]]; then
        error "This script must be run as root."
```

```
        exit 1
    fi
}

# ──────── Collectors ──────────────────────────────────────

get_os_info() {
    echo "\"os\": {"
    echo "  \"name\": \"$(uname -s)\","
    echo "  \"kernel\": \"$(uname -r)\","
    echo "  \"arch\": \"$(uname -m)\","
    echo "  \"hostname\": \"$(hostname)\","
    echo "  \"uptime\": \"$(uptime -p)\""
    echo "},"
}

get_cpu_info() {
    echo "\"cpu\": {"
    echo "  \"model\": \"$(lscpu | grep 'Model name' | cut -d ':' -f2 | xargs)\","
    echo "  \"cores\": $(nproc),"
    echo "  \"load_average\": \"$(uptime | awk -F'load average: ' '{print $2}')\""
    echo "},"
}

get_mem_info() {
```

```
    local meminfo=$(free -m | awk '/^Mem:/ {print $2","$3","$4}')
    IFS=',' read -r total used free <<< "$meminfo"
    echo "\"memory\": {"
    echo "  \"total_mb\": $total,"
    echo "  \"used_mb\": $used,"
    echo "  \"free_mb\": $free"
    echo "},"
}

get_disk_info() {
    echo "\"disk\": ["
    df -h --output=source,fstype,size,used,avail,pcent,target | tail -n +2 | awk '{
        printf "{ \"device\": \"%s\", \"type\": \"%s\", \"size\": \"%s\", \"used\": \"%s\", \"avail\": \"%s\", \"use%%\": \"%s\", \"mount\": \"%s\" },\n", $1, $2, $3, $4, $5, $6, $7
    }' | sed '$ s/,$//'
    echo "],"
}

get_users() {
    echo "\"users\": ["
    who | awk '{print "{ \"user\": \""$1"\", \"tty\": \""$2"\", \"login_time\": \""$3" "$4"\" },"}' | sed '$ s/,$//'
    echo "],"
}

get_services() {
```

```
    echo "\"failed_services\": ["
    systemctl --failed --no-pager --no-legend | awk '{print "{ \"service\": \""$1"\", \"load\": \""$2"\", \"status\": \""$3"\" },"}' | sed '$ s/,$//'
    echo "]"
}

# ──────── Report Generator ──────────────────────────────

generate_report() {
    {
        echo "{"
        get_os_info
        get_cpu_info
        get_mem_info
        get_disk_info
        get_users
        get_services
        echo "}"
    } > "$OUT_DIR/$REPORT_NAME"
    success "Report generated at $OUT_DIR/$REPORT_NAME"
}

# ──────── CLI Argument Parsing ──────────────────────────

usage() {
    echo "Usage: $0 [-q|--quiet]"
    exit 1
```

```
}

parse_args() {
    while [[ $# -gt 0 ]]; do
        case "$1" in
            -q|--quiet) QUIET=true ;;
            -h|--help) usage ;;
            *) usage ;;
        esac
        shift
    done
}

# ──────── Main ──────────────────────────────────────────────────

main() {
    parse_args "$@"
    check_root
    init_dirs
    log "Starting system diagnostics..."
    generate_report
}

main "$@"
```

Concepts Demonstrated

Feature	Description
set -euo pipefail	Pro-grade script safety
trap	Catch and report errors with context
Modular functions	Clean separation of logic
CLI flags	Quiet mode for cron jobs
df, free, lscpu, systemctl, uptime	Core system tools
JSON by hand	Simulates output usable by APIs/dev tools
Logging	Timestamped log for audit trail
Root check	Privileged enforcement

How to Use

bash

CopyEdit

```
chmod +x sysreport.sh
sudo ./sysreport.sh          # Normal mode
sudo ./sysreport.sh --quiet   # Silent/cron mode
```

Cron Job Setup Example

cron

CopyEdit

```
0 * * * * /usr/local/bin/sysreport.sh --quiet
```

Generates a fresh JSON system report every hour, silently.