"User Audit Report Generator"

technical debt

Scenario:

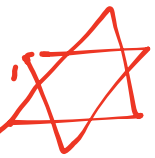
You are a system administrator tasked with writing a script that audits system users and generates a report. The script should be capable of:

1. Accepting a username or "all" as an argument.
2. For a specific user, report:
   * If the user exists.
   * Their UID, home directory, and shell.
   * Their last login (if available).
3. For "all", iterate through all local users and generate a summary report.
4. Include error handling for:
   * Missing arguments.
   * Nonexistent users.
   * Missing commands or read failures.
5. Allow optional output to a specified file.
6. Trap errors and exit cleanly.
7. Include meaningful comments and organized functions.

1) Breakdown the problem
2) Choose the commands
3) Skeleton
4) Full Script

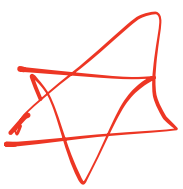
- Accept username or 'all' — read -p
- Check if user exists — id or getent passwd
- Check UID, home directory & shell — getent passwd
- Find last login — lastlog
- Iterate through users — while loop + IFS
- Handle missing args — if [[-z "$user_input"]]
- Handle Command failures — set -euo pipefail

- Handle nonexistent users — print error
- Optional output file — :-
- Trap



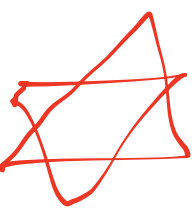
lastlog -u $user

Gives the last log in timestamp & IP address



id $user

Gives uid/groupID of the user



getent passwd $user

Gives name, uid, gid, home dir, shell

while loop

```
root@ip-172-31-27-45:~# cat script.sh
#!/bin/bash
count=1
while [[ $count -le 3 ]] ; do
        echo "Count is $count"
        ((count++))
done
```

IFS = Internal Field Separator (Delimiter)

```
#!/bin/bash
input="users.txt"

while IFS=: read -r name _ uid gid _ homedir shell; do
       echo "$homedir"
done < "$input"
~
~
```

Default value of input variable

Name = "$1"

Name = "${1:-Nithya}"

/dev/stdout

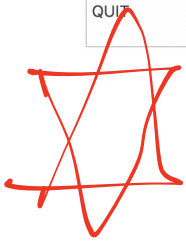
output = "${2:- /dev/stdout}"

TRAP

Catching errors

```
#!/bin/bash
trap 'echo "Interruption happened"; exit 1' INT
trap 'echo "ERROR occurred"' ERR
sleep 10
~
~
~
~
```

Signal	Name	Trigger/Event	Typical Use
EXIT	Exit	Script exits for any reason (even success)	Cleanup (delete temp files, save logs)
INT	Interrupt	**Ctrl+C** by user	Graceful interruption
TERM	Terminate	Sent by kill (default)	Clean shutdown from system/process
ERR	Error	A command fails (with set -e)	Debugging/logging failures
HUP	Hangup	Terminal closed or user logged out	Restart script or cleanup
QUIT	Quit	Ctrl+\ or signal 3	Often used for debugging core dumps



Functions

```
#!/bin/bash
greetings() { echo "Hello World "; }
greetings
~
~
```

```
#!/bin/bash
greetings() { echo "Hello $1 "; }
greetings "Veda"
~
~
```

SKELETON

```
#!/bin/bash
set -euo pipefail
trap . . . . . ERR

USER_INPUT="$1"
OUTPUT_FILE="${2:-/dev/stdout}"

print_help()

get_user_info()
    getent passwd

get_report()
{   while IFS: do
        get_user_info $usr
    done < /etc/passwd
}
```

1

2

Main Logic

Input Validation Logic

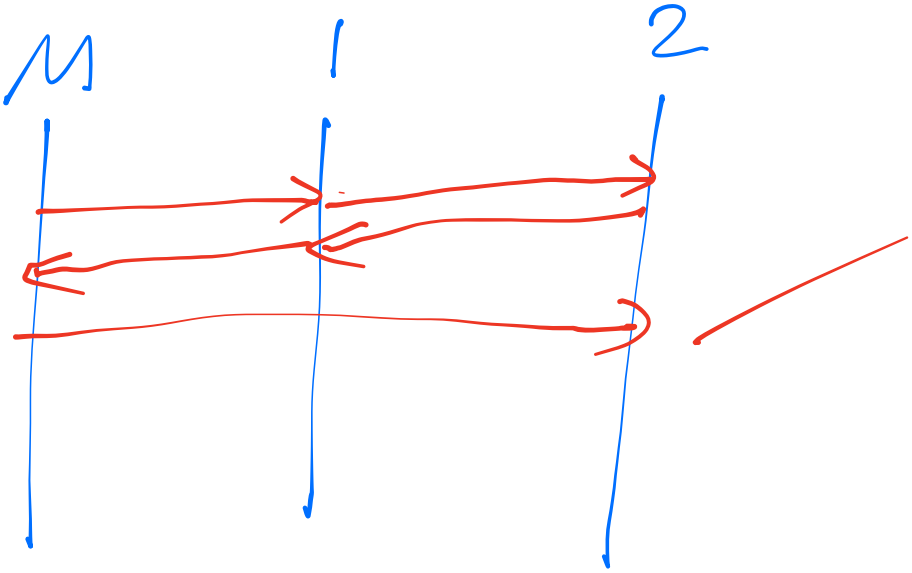
print_help()

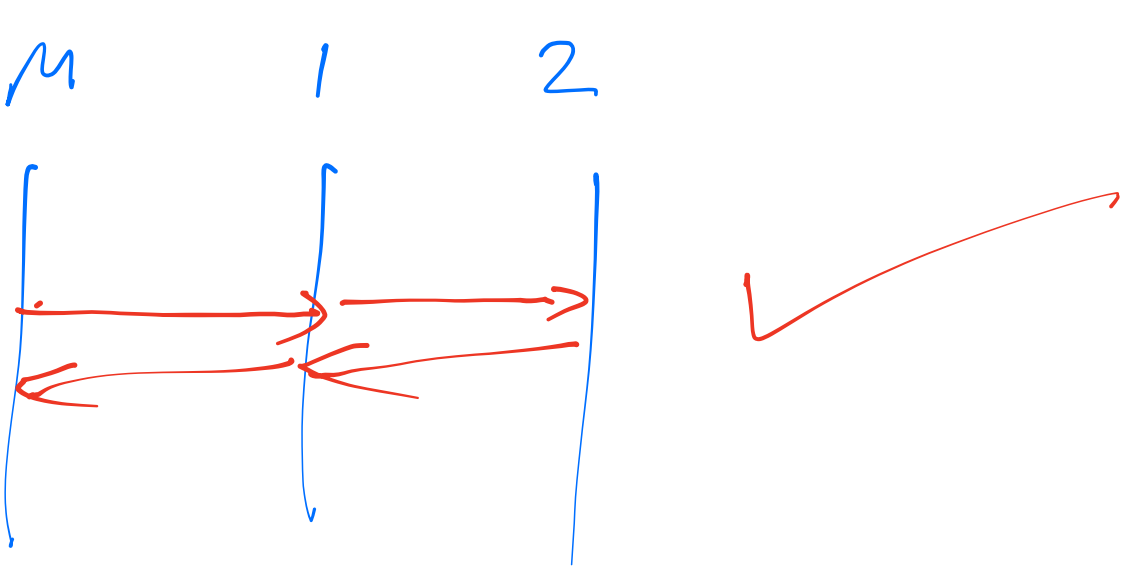
if input = "all"

get_report 2

else

get_user_info $name 1






```
get_user_info()
    getent passwd $user

get_report()
    if $1 == all ; do
        while IFS ; do
            get_user_info $name
        done > /etc/passwd
    else
        get_user_info $1
```

Main logic

get_report

```
root@ip-172-31-27-45:~# cat script.sh
#!/bin/bash
set -euo pipefail
trap 'echo "Error occurred at line $LINENO " ; exit 1' ERR
# $LINENO is inbuilt, gives line number

USER_INPUT="${1:-}"
OUTPUT_FILE="${2:-/dev/stdout}"

print_help(){
  echo "Correct usage is $0 [username|all] [outputfilename] " # $0 is inbuild, gives script name
}

get_user_info(){

  user="$1"
  if ! id "$user" &>/dev/null; then
          echo "User $user does not exist" >> "$OUTPUT_FILE"
          exit 1
  fi
  local uid shell home last_login
  uid=$(id -u $user)
  home=$(getent passwd "$user" | cut -d: -f6 )
  shell=$(getent passwd "$user" | cut -d: -f7 )
  last_login=$(lastlog -u $user | tail -n 1 )

  echo "User: $user" >> "$OUTPUT_FILE"
  echo "UID: $uid"  >> "$OUTPUT_FILE"
  echo "Home Directory: $home" >> "$OUTPUT_FILE"
  echo "Shell: $shell" >> "$OUTPUT_FILE"
  echo "Last Login: $last_login"  >> "$OUTPUT_FILE"

}

get_report(){
   if [[ "$USER_INPUT" == "all" ]]; then
      while IFS=: read -r name _ uid gid _ homedir shell ; do
              get_user_info "$name"
      done < /etc/passwd
   else
          get_user_info "$USER_INPUT"
   fi

}
```

```
if [[ -z "$USER_INPUT" ]]; then
        echo "No user input provided"
        print_help
fi

get_report
```