

LAB 3 Users Notes

Monday, October 14, 2024 12:34 PM

1. It took me around 20 to 22 hours to complete the assignment

2. Notes for the script is given below:

The bash script is designed to automate user and group management for Dunder Mifflin employees across both Linux and FreeBSD systems.

1. Group Management: The script checks for the existence of predefined groups (managers, sales, accounting). If a group does not exist, it is created using `groupadd` on Linux and `pw group add` on FreeBSD.

2. Employee Details: A hard-coded list of employees, including username, full name, password, UID, GID, home directory, shell, and job title, is used to create user accounts. Each user's home directory is created with permissions that restrict access to the owner only.

3. Assigning Groups: Users are assigned to groups based on their job titles. If a job title contains 'managers', 'sales', or 'accountant', the user is added to the corresponding group. Users with the job title 'administrator' are added to the system-level admin group (wheel or sudo).

4. Umask Configuration: The script sets the umask to 007, ensuring that new files are created with permissions accessible only by the user and their group, not by others. This configuration is applied globally by writing to `/etc/profile.d/umask.sh`.

5. Shared Directories: On machine E, the script creates shared directories for the groups (managers, sales, accounting) with strict ownership and permissions (chmod 2770). Only group members can access and modify files within these directories.

The script is designed to work in both Linux and FreeBSD.

3. The Bash Script is given below:

```
#!/bin/bash
# Define groups
groups=("managers" "sales" "accounting")
# Function to check if a group exists
group_exists() {
    local group_name="$1"
    if getent group "$group_name" > /dev/null 2>&1; then
        return 0 # Group exists
    else
        return 1 # Group does not exist
    fi
}
# Function to add user to group based on job title and system environment
add_user_to_group() {
    local group="$1"
    local username="$2"
    # Determine if 'usermod' or 'pw group mod' should be used
    if command -v usermod && /dev/null; then
        usermod -aG "$group" "$username"
    else
        pw group mod "$group" -m "$username"
    fi
    echo "$username added to $group group."
```

```

}
# Function to set umask in /etc/profile.d/umask.sh
set_umask() {
    local umask_file="/etc/profile.d/umask.sh"
    local umask_setting="umask 007"
    # Create or overwrite the umask file with the desired setting
    echo "# Set default umask" > "$umask_file"
    echo "$umask_setting" >> "$umask_file"
    echo "Umask set to 007 in $umask_file. Please log out and log back in to apply the changes."
}
# Function to assign group based on job title
assign_group_based_on_job() {
    local job_title="$1"
    local username="$2"
    # Add user to groups based on their job title
    if [[ "$job_title" == *"Manager"* ]]; then
        add_user_to_group "managers" "$username"
    elif [[ "$job_title" == *"Sales"* ]]; then
        add_user_to_group "sales" "$username"
    elif [[ "$job_title" == *"Accountant"* ]]; then
        add_user_to_group "accounting" "$username"
    elif [[ "$job_title" == *"Administrator"* ]]; then
        if group_exists "wheel"; then
            add_user_to_group "wheel" "$username"
        elif group_exists "sudo"; then
            add_user_to_group "sudo" "$username"
        fi
    else
        echo "$username does not match any predefined groups."
    fi
}
create_shared_directory() {
    group=$1
    dir="/home/$group"
    if [ ! -d "$dir" ]; then
        mkdir "$dir"
        echo "Created shared directory for group $group at $dir"
    else
        echo "Shared directory already exists with $group at $dir"
    fi
    chown :"$group" "$dir"
    chmod 2770 "$dir"
    echo "Set permissions for shared directory $dir"
}
# Create groups if they don't exist
for group in "${groups[@]}"; do
    if command -v groupadd &> /dev/null; then
        if ! getent group "$group" > /dev/null 2>&1; then
            groupadd "$group"
            echo "Group $group created."
        else
            echo "Group $group already exists."
        fi
    else
        if ! getent group "$group" > /dev/null 2>&1; then
            pw group add "$group"
            echo "Group $group created."
        else
            echo "Group $group already exists."
        fi
    fi
done

```

```

done
users=(
    "mscott|Michael Scott|password|3001|3001|/bin/bash|/home/mscott|Regional Manager"
    "jhalpert|Jim Halpert|password|3002|3002|/bin/bash|/home/jhalpert|Assistant Regional Manager"
    "dschrute|Dwight Schrute|password|3003|3003|/bin/bash|/home/dschrute|Assistant to the
Regional Manager"
    "pbeesly|Pam Beesly|password|3004|3004|/bin/bash|/home/pbeesly|Receptionist"
    "abernard|Andy Bernard|password|3005|3005|/bin/bash|/home/abernard|Regional Director in
Charge of Sales"
    "plapin|Phyllis Lapin|password|3006|3006|/bin/bash|/home/plapin|Sales Representative"
    "shudson|Stanley Hudson|password|3007|3007|/bin/bash|/home/shudson|Sales Representative"
    "amartin|Angela Martin|password|3008|3008|/bin/bash|/home/amartin|Senior Accountant"
    "omartinez|Oscar Martinez|password|3009|3009|/bin/bash|/home/omartinez|Accountant"
    "kmalone|Kevin Malone|password|3010|3010|/bin/bash|/home/kmalone|Accountant"
    "dphilbin|Darryl Philbin|password|3011|3011|/bin/bash|/home/dphilbin|Warehouse Foreman"
    "k Kapoor|Kelly Kapoor|password|3012|3012|/bin/bash|/home/k Kapoor|Customer Service
Representative"
    "tflenderson|Toby Flenderson|password|3013|3013|/bin/bash|/home/tflenderson|Human
Resources"
    "mpalmer|Meredith Palmer|password|3014|3014|/bin/bash|/home/mpalmer|Supplier Relations"
    "cbratton|Creed Bratton|password|3015|3015|/bin/bash|/home/cbratton|Quality Assurance
Representative"
    "arma2040|Aravindh Goutham|password|3016|3016|/bin/bash|/home/arma2040|System
Administrator"
)
for user in "${users[@]}"; do
    IFS='|' read -r username gecos password uid gid shell home job_title <<< "$user"

    if command -v useradd &> /dev/null; then
        # Create the user with specified details
        useradd -m -u "$uid" -s "$shell" -c "$gecos" "$username"
        echo "$username:$password" | chpasswd
    else
        # Create the user with specified details
        pw user add "$username" -m -u "$uid" -s "$shell" -c "$gecos"
        echo "$password" | pw usermod "$username" -h 0
    fi
    assign_group_based_on_job "$job_title" "$username"
    echo "User $username ($gecos) created with UID $uid, GID $gid, and home directory $home."
    # Set the password for the user
    chmod 700 "$home"

done
if [ "$(hostname)" == "machinee.dundermifflin.com" ]; then
    for group in "${groups[@]}"; do
        create_shared_directory "$group"
    done
fi
# Set the umask for the new files and directories
set_umask

```