

Configuration Management Script BASH

Monday, October 14, 2024 12:12 PM

```
#!/bin/bash
```

```
# Function to get content from /etc/passwd
```

```
get_passwd_entries() {  
    cat /etc/passwd  
}
```

```
# Function to get password hashes from /etc/shadow (requires sudo)
```

```
get_shadow_passwords() {  
    sudo cat /etc/shadow  
}
```

```
# Function to get groups for users from /etc/group
```

```
get_user_groups() {  
    cat /etc/group  
}
```

```
# Function to create a task for creating users in the playbook
```

```
create_user_task() {  
    local username="$1"  
    local uid="$2"  
    local gid="$3"  
    local gecosc="$4"  
    local home="$5"  
    local shell="$6"  
    local password_hash="$7"  
    local groups="$8"  
  
    echo " - name: Ensure user $username exists"  
    echo "   ansible.builtin.user:"  
    echo "     name: $username"  
    echo "     uid: $uid"  
    echo "     group: $gid"  
    echo "     comment: \"$gecos\""  
    echo "     shell: $shell"  
    echo "     password: $password_hash"  
    echo "     home: $home"  
    echo "     groups: $groups"  
    echo "     state: present"  
}
```

```
# Function to create a task for ensuring groups exist
```

```
create_group_task() {  
    local group_name="$1"  
    echo " - name: Ensure group $group_name exists"  
    echo "   ansible.builtin.group:"  
    echo "     name: $group_name"  
    echo "     state: present"
```

```
}
```

```
# Function to create tasks for wheel and sudo groups
```

```
create_wheel_group_task() {
```

```
    local username="$1"
```

```
    local uid="$2"
```

```
    local gid="$3"
```

```
    local geccos="$4"
```

```
    local home="$5"
```

```
    local shell="$6"
```

```
    local password_hash="$7"
```

```
    local groups="$8"
```

```
    echo "  - name: Check if wheel group exists"
```

```
    echo "    ansible.builtin.getent:"
```

```
    echo "      database: group"
```

```
    echo "      key: wheel"
```

```
    echo "      fail_key: false"
```

```
    echo "    register: wheel_group"
```

```
    echo ""
```

```
    echo "  - name: Check if sudo group exists"
```

```
    echo "    ansible.builtin.getent:"
```

```
    echo "      database: group"
```

```
    echo "      key: sudo"
```

```
    echo "      fail_key: false"
```

```
    echo "    register: sudo_group"
```

```
    echo ""
```

```
    echo "  - name: Ensure user $username is part of the wheel group"
```

```
    echo "    ansible.builtin.user:"
```

```
    echo "      name: $username"
```

```
    echo "      uid: $uid"
```

```
    echo "      group: $gid"
```

```
    echo "      comment: \"$geccos\""
```

```
    echo "      home: $home"
```

```
    echo "      shell: $shell"
```

```
    echo "      password: \"$password_hash\""
```

```
    echo "      groups: \"wheel\"${ [ -n \"$groups\" ] && echo \", $groups\" }"
```

```
    echo "      state: present"
```

```
    echo "      append: yes"
```

```
    echo "    when: wheel_group.ansible_facts.getent_group.wheel is not none"
```

```
    echo ""
```

```
    echo "  - name: Ensure user $username is part of the sudo group"
```

```
    echo "    ansible.builtin.user:"
```

```
    echo "      name: $username"
```

```
    echo "      uid: $uid"
```

```
    echo "      group: $gid"
```

```
    echo "      comment: \"$geccos\""
```

```
    echo "      home: $home"
```

```
    echo "      shell: $shell"
```

```
    echo "      password: \"$password_hash\""
```

```
    echo "      groups: \"sudo\"${ [ -n \"$groups\" ] && echo \", $groups\" }"
```

```
    echo "      state: present"
```

```
    echo "      append: yes"
```

```

    echo "    when: sudo_group.ansible_facts.getent_group.sudo is not none"
}

# Main function to generate the playbook
main() {
    # String to hold the playbook content
    playbook_content=""
- hosts: all
  become: true
  tasks:
"

    # Groups to create
    groups_to_create=("managers" "accounting" "sales")
    for group in "${groups_to_create[@]"; do
        playbook_content+=$(create_group_task "$group")
        playbook_content+=$'\n'
    done

    # Get passwd entries
    passwd_entries=$(get_passwd_entries)

    # Get shadow passwords
    shadow_data=$(get_shadow_passwords)

    # Parse shadow data into an associative array
    declare -A shadow_dict
    while IFS=: read -r username password_hash _; do
        shadow_dict["$username"]="$password_hash"
    done <<< "$shadow_data"

    # Parse group data into an associative array
    declare -A group_dict
    while IFS=: read -r group_name _ gid members; do
        IFS=',' read -ra users <<< "$members"
        for user in "${users[@]"; do
            group_dict["$user"]+="$group_name,"
        done
    done <<< "$(get_user_groups)"

    # Loop through passwd entries and create tasks for users with home directories
    while IFS=: read -r username _ uid gid gecos home shell; do
        if [[ "$home" == /home/* ]]; then
            password_hash="${shadow_dict[$username]:-}" # Default to "" if no entry in shadow
            groups="${group_dict[$username]%,}" # Remove trailing comma

            # Create the task for user creation
            if [[ "$groups" == "wheel" ]]; then
                groups="${groups//wheel/}"
                playbook_content+=$(create_wheel_group_task "$username" "$uid" "$gid" "$gecos" "$home"
"$shell" "$password_hash" "$groups")
            else
                playbook_content+=$(create_user_task "$username" "$uid" "$gid" "$gecos" "$home" "$shell"

```

```
"$password_hash" "$groups")
    fi
    playbook_content+=$'\n'
    fi
done <<< "$passwd_entries"

# Save the playbook to a YAML file
filename="dmusers.yaml"
echo "$playbook_content" > "$filename"

# Change owner of the file
chown arma2040 "$filename"

echo "Ansible playbook '$filename' has been created."
}

# Entry point
main
```