CW2 Advanced User Account Management Worksheet

This worksheet is designed to help you complete the User Account Management part of your coursework. You will demonstrate your ability to create, manage, and document user accounts on Linux, Windows, and Windows Server systems, using both Command-Line Interface (CLI) and Graphical User Interface (GUI) tools. Higher-level submissions should incorporate principles of Role-Based Access Control (RBAC), appropriate privilege assignment, and password policy enforcement.

# 1. Plan Your User Accounts

Your first task is to determine the user roles needed for a small tech company. Think about the types of users who might need access to the system, and their required privileges.

- Admin (full system control)

- Developer (access to coding tools and limited admin rights)

- Intern (read-only access to shared folders)

- HR (access to employee records, limited to a folder)

❓ Task: Create a table listing the usernames, their roles, required access levels, and notes about which OS each user should exist on.

# 2. Creating Users Using GUI Tools

Use the graphical tools available in each operating system to create the planned user accounts.

Include screenshots of each user account you create.

# 3. Creating Users Using CLI Tools

Use appropriate terminal or PowerShell commands to replicate the above users using the CLI.

Linux Commands:

- Create user: `sudo adduser username`

- Add to sudo group: `sudo usermod -aG sudo username`

- Lock user: `sudo usermod -L username`

- Password policy: `sudo chage -M 30 username`

Windows Commands:

- Create user: `net user username password /add`

- Add to Administrators: `net localgroup administrators username /add`

- List users: `net user`

- Set password expiry: `wmic useraccount where name='username' set PasswordExpires=True`

Windows Server (PowerShell):

- New user: `New-LocalUser -Name 'username' -Password (ConvertTo-SecureString 'P@ssword123' -AsPlainText -Force)`

- Add to group: `Add-LocalGroupMember -Group 'Administrators' -Member 'username'`

- Set password expiry: `Set-LocalUser -Name 'username' -PasswordNeverExpires $False`

# 4. Implementing Role-Based Access Control (RBAC)

Assign users to groups based on their roles, then set permissions on files/folders accordingly.

❓ Task: Create three roles (e.g., Admins, Developers, Interns). Add users to appropriate groups and restrict folder access accordingly. Document commands and include screenshots.

# 5. Reflection: CLI vs GUI

❓ Task: Which did you find easier and why? What are the pros and cons of each method for system administrators?