

## User Management Commands:

1. Create User: `useradd username`
2. Set Password: `passwd username`
3. Delete User: `userdel username`
4. Modify User Details: `usermod -option value username`
5. Assign User to Group: `usermod -aG groupname username`
6. Remove User from Group: `gpasswd -d username groupname`
7. List Users: `cut -d: -f1 /etc/passwd`
8. List User Details: `id username`
9. Lock User Account: `passwd -l username`
10. Unlock User Account: `passwd -u username`
11. Display Last Login: `last username`
12. Set User Expiry: `chage -E YYYY-MM-DD username`
13. Force User to Change Password on Next Login: `chage -d 0 username`
14. Display User's Groups: `groups username`
15. Grant Sudo Access: `usermod -aG sudo username`
16. Revoke Sudo Access: `deluser username sudo`
17. Show Users Logged In: `who`
18. Show Currently Logged-In Users: `who -u`
19. Kill User's Active Session: `pkill -KILL -u username`
20. Assign Home Directory: `usermod -m -d /path/to/directory username`
21. Change Default Shell: `chsh -s /bin/bash username`
22. Display User Quotas: `repquota -a`
23. Limit User Logins: `usermod -L username`
24. Allow User Logins: `usermod -U username`
25. Set Default User Skeleton: `useradd -D -b /path/to/skeleton`

26. View User's Login Shell: `echo $SHELL`
27. Set User Environment Variables: `export VARNAME=value`
28. Display Expired Users: `chage -l username`
29. Create Group: `groupadd groupname`
30. Delete Group: `groupdel groupname`
31. Modify Group Name: `groupmod -n newgroupname oldgroupname`
32. List Groups: `cut -d: -f1 /etc/group`
33. Display Group Details: `getent group groupname`
34. Assign Primary Group: `usermod -g groupname username`
35. Change Group Ownership of a File: `chgrp groupname filename`
36. Set Group ID (SGID) on a Directory: `chmod g+s directory`
37. Set Default File Creation Group: `umask 002`
38. List User Processes: `ps -u username`
39. Show System Users: `awk -F: '$3 >= 1000 && $1 != "nobody" {print $1}'`  
`/etc/passwd`
40. Display User Activity: `ac -d`
41. Set Default Password Aging Policy: `chage -m MIN_DAYS -M MAX_DAYS -I INACTIVE_DAYS -W WARN_DAYS username`
42. Restrict Users from Changing Password: `chage -M 99999 username`
43. Change Password Aging Information: `chage -l username`
44. Display Group Memberships: `getent passwd | grep -E`  
`':(username|groupname):'`
45. Show Group Password Aging Policy: `chage -l groupname`
46. Limit Simultaneous User Logins: `usermod -L -f 0 username`
47. Allow Simultaneous User Logins: `usermod -U -f -1 username`
48. Display Group Password Details: `getent group groupname`
49. Create User with Specific UID: `useradd -u UID username`
50. Set User's Default Login Shell: `usermod -s /path/to/shell username`
51. Display Current User's Groups:

groups

52. Change User's Primary Group:

newgrp groupname

53. Change Group Password:

gpasswd groupname

54. List Group Members:

getent group groupname

55. List All Users and Their Details:

getent passwd

56. List All Groups and Their Members:

getent group

57. Set User's Login Shell:

chsh -s /path/to/shell username

58. Set User's Real Name:

`chfn -f "Full Name" username`

59. Show Files Owned by User:

`find / -user username`

60. Show Files Owned by Group:

`find / -group groupname`

61. Display User Resource Limits:

`ulimit -a`

62. Change User's Home Directory:

`usermod -d /new/path/to/home username`

63. Change Default User Shell for All Users:

`sed -i 's/oldshell/newshell/' /etc/passwd`

64. Limit Core Dumps for User:

`ulimit -c 0`

65. View User's Login History:

lastlog username

66. Display Last Failed Login Attempts:

grep -i "failed" /var/log/auth.log

67. Show Expired Passwords:

chage -l | grep "Password expires"

68. Set Maximum Password Age:

chage -M MAX\_DAYS username

69. Set Minimum Password Age:

chage -m MIN\_DAYS username

70. Set Warning Days before Password Expiry:

chage -W WARN\_DAYS username

71. Display Last Password Change Date:

chage -l username | grep "Last password change"

72. Check Password Complexity Requirements:

`pam_pwquality --test`

73. Lock User Account After Failed Login Attempts:

`pam_tally2 -u username -l`

74. Unlock User Account:

`pam_tally2 -u username -r`

75. Display User's Privileges:

`sudo -l -U username`

76. List Sudo Rules:

`sudo grep -E '^[^#]*\s+\w+\s+= ' /etc/sudoers /etc/sudoers.d/*`

77. Change Group Ownership Recursively:

`chown -R groupname /path/to/directory`

78. Add User to Multiple Groups:

`usermod -aG group1,group2,group3 username`

79. Check User's Login Shell:

echo \$SHELL

80. Display User's Default Umask:

umask

81. Set Default Umask:

umask new\_umask

82. List Groups a User is a Member of:

groups username

83. Remove User from Multiple Groups:

userdel -G group1,group2,group3 username

84. Set User's Password Expiry Date:

chage -M MAX\_DAYS -I INACTIVE\_DAYS username

85. Display System Users:

awk -F: '\$3 < 1000 {print \$1}' /etc/passwd

86. Show System Groups:

```
awk -F: '$3 < 1000 {print $1}' /etc/group
```

87. Create User with Specific GID:

```
useradd -g GID username
```

88. Set User's GID:

```
usermod -g GID username
```

89. Set GID for New Files in a Directory:

```
chmod g+s /path/to/directory
```

90. List Sudo Rules for a Specific User:

```
sudo -l -U username
```

91. Display Home Directory Permissions:

```
ls -ld /home/username
```

92. List All Users in a Group:

```
getent passwd | awk -F: -v group="groupname" '$4 == group {print $1}'
```

93. Add User to Sudoers File:



```
echo "username ALL=(ALL) ALL" | sudo tee -a /etc/sudoers
```

94. Remove User from Sudoers File:

```
sudo visudo
```

95. Display Account Expiry Information:

```
chage -l username
```

96. Check if a User Exists:

```
id username
```

97. Check if a Group Exists:

```
getent group groupname
```

98. Display User Login Records:

```
last username
```

99. Show Processes Owned by User:

```
ps -u username
```

100. Show Total Disk Usage for a User:

```
bash du -sh /home/username
```

Certainly! Here are more user management commands:

101. **\*\*Display User's Current Shell:\*\***

```
```bash
echo $SHELL
```
```

102. **\*\*Set User's Login Shell:\*\***

```
```bash
chsh -s /path/to/shell username
```
```

103. **\*\*List Users' Home Directories:\*\***

```
```bash
awk -F: '{print $6}' /etc/passwd
```
```

104. **\*\*Display User's Home Directory Size:\*\***

```
```bash
du -sh /home/username
```
```

105. **\*\*Create User with Custom Expiry Date:\*\***

```
```bash
useradd -e YYYY-MM-DD username
```
```

106. **\*\*Display Files Modified by User:\*\***

```
` `` bash  
  
find / -user username -exec ls -l {} \;  
` ``
```

107. **\*\*Set User's UID:\*\***

```
` `` bash  
  
usermod -u UID username  
` ``
```

108. **\*\*Change Group Ownership of User's Files:\*\***

```
` `` bash  
  
find /path/to/files -user username -exec chown newgroup '{}' \;  
` ``
```

109. **\*\*Display User's Last Password Change Time:\*\***

```
` `` bash  
  
chage -l username | grep "Last password change"  
` ``
```

110. **\*\*List All Users and Their Groups:\*\***

```
` `` bash  
  
getent passwd | awk -F: '{print $1, $4}'  
` ``
```

111. **\*\*Display Group Password Details:\*\***

```
` `` bash
```

```
getent group groupname
```

```
```\n
```

112. **\*\*Modify User's GECOS (General Electric Comprehensive Operating System) Information:\*\***

```
```bash
```

```
usermod -c "New GECOS" username
```

```
```\n
```

113. **\*\*Show Files Owned by User and Larger Than a Specific Size:\*\***

```
```bash
```

```
find / -user username -size +1M -exec ls -lh {} \;
```

```
```\n
```

114. **\*\*Set User's Default umask:\*\***

```
```bash
```

```
echo "umask new_umask" >> /etc/bash.bashrc
```

```
```\n
```

115. **\*\*Display User Processes and Resource Usage:\*\***

```
```bash
```

```
ps -u username -o pid,%cpu,%mem,cmd
```

```
```\n
```

116. **\*\*Create User and Assign Multiple Secondary Groups:\*\***

```
```bash
```

```
useradd -G group1,group2,group3 username
```

```
```\n
```

117. **\*\*Display User's Most Recently Used Files:\*\***

```
```bash
```

```
find /home/username -type f -atime -7
```

```
```
```

118. **\*\*Set User's Default umask in Profile:\*\***

```
```bash
```

```
echo "umask new_umask" >> /home/username/.bashrc
```

```
```
```

119. **\*\*Display User's Group Membership Details:\*\***

```
```bash
```

```
id username
```

```
```
```

120. **\*\*Change User's Password Non-Interactively:\*\***

```
```bash
```

```
echo "newpassword" | passwd --stdin username
```

```
```
```

121. **\*\*Display Total Disk Usage for a User:\*\***

```
```bash
```

```
du -sh /home/username
```

```
```
```

122. **\*\*Set User's Account to Expire:\*\***

```
```bash
```

```
usermod -e YYYY-MM-DD username
```

```
```\n
```

123. **\*\*Set User's Login Shell Interactively:\*\***

```
```\nbash
```

```
chsh username
```

```
```\n
```

124. **\*\*Display Users with Expiring Passwords:\*\***

```
```\nbash
```

```
chage -l | grep "Password expires" | awk '$NF < 7'
```

```
```\n
```

125. **\*\*Change User's Primary Group Interactively:\*\***

```
```\nbash
```

```
newgrp groupname
```

```
```\n
```

126. **\*\*Show Processes Running as a Specific User:\*\***

```
```\nbash
```

```
ps -U username
```

```
```\n
```

127. **\*\*Display User's Last Login Details:\*\***

```
```\nbash
```

```
lastlog -u username
```

```
```\n
```

128. **\*\*Check if a User is Logged In:\*\***

```
```bash
who | grep -w username
```
```

129. **\*\*Show Group Password Details:\*\***

```
```bash
getent group groupname
```
```

130. **\*\*Create User with a Specific Shell:\*\***

```
```bash
useradd -s /path/to/shell username
```
```

131. **\*\*Change User's Home Directory Interactively:\*\***

```
```bash
usermod -d /new/path/to/home username
```
```

132. **\*\*Display Last Login Details for All Users:\*\***

```
```bash
lastlog
```
```

133. **\*\*Display Users with No Password:\*\***

```
```bash
awk -F: '$2 == "" {print $1}' /etc/shadow
```

```

134. **\*\*Set User's Password Non-Interactively:\*\***

```
```bash
echo "username:newpassword" | chpasswd
```
```

135. **\*\*List All Users in Multiple Groups:\*\***

```
```bash
getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```
```

136. **\*\*Display User's Current Group Memberships:\*\***

```
```bash
id -Gn username
```
```

137. **\*\*Create User with Custom Home Directory:\*\***

```
```bash
useradd -m -d /path/to/custom/home username
```
```

138. **\*\*Display Users with UID Less Than 1000:\*\***

```
```bash
awk -F: '$3 < 1000 {print $1}' /etc/passwd
```
```

139. **\*\*Set Password Expiry Warning Days:\*\***



```
```bash
```

```
chage -W WARN_DAYS username
```

```
```
```

#### 140. **\*\*Display Number of Failed Login Attempts:\*\***

```
```bash
```

```
pam_tally2 -u username
```

```
```
```

#### 141. **\*\*Create User with UID and GID:\*\***

```
```bash
```

```
useradd -u UID -g GID username
```

```
```
```

#### 142. **\*\*Set GID for New Files in a Directory Interactively:\*\***

```
```bash
```

```
chmod g+s directory
```

```
```
```

#### 143. **\*\*List Users with No Shell Access:\*\***

```
```bash
```

```
awk -F: '$NF !~ "/bin/(sh|bash)" {print $1}' /etc/passwd
```

```
```
```

#### 144. **\*\*Display User's Last Password Change Time in Epoch Format:\*\***

```
```bash
```

```
chage -l username | grep "Last password change" | awk '{print $5}'
```

```
```
```

145. **\*\*List Users with UID Greater Than 1000:\*\***

```
` `` bash  
awk -F: '$3 >= 1000 {print $1}' /etc/passwd  
` ``
```

146. **\*\*Display User's Full Name:\*\***

```
` `` bash  
finger username  
` ``
```

147. **\*\*List Users with No Home Directory:\*\***

```
` `` bash  
awk -F: '$6 == "" {print $1}' /etc/passwd  
` ``
```

148. **\*\*Set Default Password Aging Policy Interactively:\*\***

```
` `` bash  
chage username  
` ``
```

149. **\*\*Show Users with No Shell Access in /etc/passwd:\*\***

```
` `` bash  
awk -F: '$7 !~ "/bin/(sh|bash)" {print $1}' /etc/passwd
```

150. Display User's Last Password Change in Human-Readable Format:

```
bash chage -l username | grep "Last password change" | awk -F: '{print $2}'
```

151. Set User's Shell to nologin:

bash usermod -s /usr/sbin/nologin username

152. Display User's Group Membership Details:

bash id -Gn username

153. Set User's Password Expiry Date:

bash chage -M MAX\_DAYS -I INACTIVE\_DAYS username

154. Show Users with No Login Shell in /etc/passwd:

bash awk -F: '\$7 == "/sbin/nologin" {print \$1}' /etc/passwd

155. Display Users in Secondary Groups:

bash getent passwd | awk -F: '{print \$1, \$4}' | grep "groupname"

156. Set Default Password Aging Policy in Days:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS username

157. List Users with Expired Passwords:

bash chage -l | grep "Password expires" | awk '\$NF < 0'

158. Show Users in Group Recursively:

bash find / -group groupname -exec ls -l {} \;

159. Set User's Default Group Interactively:

bash usermod -g groupname username

160. Check if a Group is Empty:

bash getent passwd | awk -F: -v group="groupname" '\$4 == group {print \$1}' | wc -l

161. Set Default GID for New Files:

bash umask -g GID

162. Display Users with No Password Aging:

bash awk -F: '{print \$1}' /etc/passwd | xargs -l {} chage -l {} | grep "Password expires: never"

163. Show Users in a Group with Details:

bash getent passwd | awk -F: -v group="groupname" '\$4 == group {print \$1, \$3, \$6}'

164. Set Default Shell for New Users:

bash useradd -D -s /path/to/shell

165. Display Groups a User is Not a Member of:

```
bash getent group | awk -F: -v user="username" '$NF !~ user {print $1}'
```

166. Set Maximum Inactivity Days for User:

```
bash chage -I INACTIVE_DAYS username
```

167. Display Number of Users Logged In:

```
bash who | wc -l
```

168. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

169. Set User's Login Shell to Bash:

```
bash usermod -s /bin/bash username
```

170. Show Users with No Valid Shell:

```
bash awk -F: '$NF !~ "/bin/(sh|bash)" {print $1}' /etc/passwd
```

171. Set User's Home Directory Permissions:

```
bash chmod 700 /home/username
```

172. List Users with Specific UID Range:

```
bash awk -F: '$3 >= MIN_UID && $3 <= MAX_UID {print $1}' /etc/passwd
```

173. Display System Users with No Login Shell:

```
bash awk -F: '$3 < 1000 && $7 == "/sbin/nologin" {print $1}' /etc/passwd
```

174. Set User's Shell to nologin Interactively:

```
bash chsh -s /sbin/nologin username
```

175. List Users Excluded from Password Aging:

```
bash awk -F: '$8 == "!" {print $1}' /etc/shadow
```

176. Show Users with No Home Directory Permissions:

```
bash find /home -maxdepth 1 -type d -not -executable
```

177. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

178. List Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

179. Display Users with No Shell Access in /etc/passwd:

```
bash awk -F: '$7 !~ "/bin/(sh|bash)" {print $1}' /etc/passwd
```

180. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

181. Show Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

182. Set Group Password Expiry Date:

```
bash chage -M MAX_DAYS groupname
```

183. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

184. Set Password Complexity Requirements:

```
bash pam_pwquality --test
```

185. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

186. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

187. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

188. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

189. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

190. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

191. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

192. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

193. Display Users with No Login Shell in /etc/passwd:

bash awk -F: '\$7 == "/sbin/nologin" {print \$1}' /etc/passwd

194. Set Default Shell for All Users Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

195. Show Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

196. Set Default Password Aging Policy Non-Interactively:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS -I INACTIVE\_DAYS  
username

197. Display Users with UID and GID Mismatch:

bash awk -F: '\$3 != \$4 {print \$1}' /etc/passwd

198. Set Default Password Aging Policy for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS -E  
EXPIRY\_DATE username

199. List Users with No Password Aging:

bash awk -F: '\$8 == "" {print \$1}' /etc/shadow

200. Set Maximum Inactivity Days for Group:

bash chage -I INACTIVE\_DAYS groupname

201. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

202. Set Password Expiry Warning Days for Group:

bash chage -W WARN\_DAYS groupname

203. List Users with Specific Shell:

bash awk -F: -v shell="/path/to/shell" '\$NF == shell {print \$1}' /etc/passwd

204. Set Default Password Aging Policy in Days for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS username

205. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

206. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

207. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

208. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

209. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

210. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

211. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

212. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

213. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

214. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

215. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

216. Set Default Shell for All Users Non-Interactively:

```
```bash  
sed -i 's/oldshell/newshell/' /etc/passwd  
``
```

217. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

218. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

219. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

220. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

221. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

222. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

223. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

224. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

225. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

226. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

227. Display Users with UID and GID Mismatch:

```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

228. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E  
EXPIRY_DATE username
```

229. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

230. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

231. Display Users with UID and GID Matching:



bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

232. Set Password Expiry Warning Days for Group:

bash chage -W WARN\_DAYS groupname

233. List Users with Specific Shell:

bash awk -F: -v shell="/path/to/shell" '\$NF == shell {print \$1}' /etc/passwd

234. Set Default Password Aging Policy in Days for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS username

235. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

236. Set Default Password Aging Policy for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS  
groupname

237. List Users with Expired Accounts:

bash awk -F: '{print \$1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"

238. Set Group Password Aging Policy:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS groupname

239. Display Users with No Home Directory Ownership:

bash find /home -maxdepth 1 -type d -not -user root

240. Set Default Shell for All Users Interactively:

bash chsh -s /path/to/shell

241. List Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

242. Set Default Password Aging Policy Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

243. Display Users with No Home Directory:

bash awk -F: '\$6 == "" {print \$1}' /etc/passwd

244. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

245. Display Users with No Login Shell in /etc/passwd:

```
```bash
```

```
awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

246. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

247. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

248. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

249. Display Users with UID and GID Mismatch:

```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

250. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E  
EXPIRY_DATE username
```

251. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

252. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

253. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

254. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

255. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

256. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

257. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

258. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

259. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

260. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

261. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

262. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

263. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

264. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

265. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

266. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

267. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

268. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

269. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

270. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

271. Display Users with UID and GID Mismatch:

```
```bash
```

```
awk -F: '$3 != $4 {print $1}' /etc/passwd
```

272. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E  
EXPIRY_DATE username
```

273. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

274. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

275. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

276. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

277. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

278. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

279. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

280. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

281. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

282. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

283. Display Users with No Home Directory Ownership:

bash find /home -maxdepth 1 -type d -not -user root

284. Set Default Shell for All Users Interactively:

bash chsh -s /path/to/shell

285. List Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

286. Set Default Password Aging Policy Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

287. Display Users with No Home Directory:

bash awk -F: '\$6 == "" {print \$1}' /etc/passwd

288. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

289. Display Users with No Login Shell in /etc/passwd:

bash awk -F: '\$7 == "/sbin/nologin" {print \$1}' /etc/passwd

290. Set Default Shell for All Users Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

291. Show Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

292. Set Default Password Aging Policy Non-Interactively:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS -I INACTIVE\_DAYS  
username

293. Display Users with UID and GID Mismatch:

bash awk -F: '\$3 != \$4 {print \$1}' /etc/passwd

294. Set Default Password Aging Policy for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS -E  
EXPIRY\_DATE username

295. List Users with No Password Aging:

bash awk -F: '\$8 == "" {print \$1}' /etc/shadow

296. Set Maximum Inactivity Days for Group:

bash chage -I INACTIVE\_DAYS groupname

297. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

298. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

299. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

300. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

301. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

302. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

303. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

304. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

305. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

306. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

307. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

308. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

309. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

310. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

311. Display Users with No Login Shell in /etc/passwd:

bash awk -F: '\$7 == "/sbin/nologin" {print \$1}' /etc/passwd

312. Set Default Shell for All Users Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

313. Show Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

314. Set Default Password Aging Policy Non-Interactively:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS -I INACTIVE\_DAYS  
username

315. Display Users with UID and GID Mismatch:

bash awk -F: '\$3 != \$4 {print \$1}' /etc/passwd

316. Set Default Password Aging Policy for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS -E  
EXPIRY\_DATE username

317. List Users with No Password Aging:

bash awk -F: '\$8 == "" {print \$1}' /etc/shadow

318. Set Maximum Inactivity Days for Group:

bash chage -I INACTIVE\_DAYS groupname

319. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

320. Set Password Expiry Warning Days for Group:

bash chage -W WARN\_DAYS groupname

321. List Users with Specific Shell:

bash awk -F: -v shell="/path/to/shell" '\$NF == shell {print \$1}' /etc/passwd

322. Set Default Password Aging Policy in Days for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS username

323. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

324. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

325. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

326. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

327. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

328. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

329. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

330. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

331. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

332. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

333. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

334. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

335. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

336. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

337. Display Users with UID and GID Mismatch:



```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

338. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E EXPIRY_DATE username
```

339. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

340. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

341. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

342. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

343. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

344. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

345. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

346. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS groupname
```

347. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

348. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

349. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

350. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

351. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

352. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

353. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

354. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

355. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

356. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

357. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

358. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

359. Display Users with UID and GID Mismatch:

```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

360. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E  
EXPIRY_DATE username
```

361. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

362. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

363. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

364. Set Password Expiry Warning Days for Group:

bash chage -W WARN\_DAYS groupname

365. List Users with Specific Shell:

bash awk -F: -v shell="/path/to/shell" '\$NF == shell {print \$1}' /etc/passwd

366. Set Default Password Aging Policy in Days for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS username

367. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

368. Set Default Password Aging Policy for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS  
groupname

369. List Users with Expired Accounts:

bash awk -F: '{print \$1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"

370. Set Group Password Aging Policy:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS groupname

371. Display Users with No Home Directory Ownership:

bash find /home -maxdepth 1 -type d -not -user root

372. Set Default Shell for All Users Interactively:

bash chsh -s /path/to/shell

373. List Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

374. Set Default Password Aging Policy Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

375. Display Users with No Home Directory:

bash awk -F: '\$6 == "" {print \$1}' /etc/passwd

376. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

377. Display Users with No Login Shell in /etc/passwd:

```bash

```
awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

..

378. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

379. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

380. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

381. Display Users with UID and GID Mismatch:

```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

382. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E  
EXPIRY_DATE username
```

383. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

384. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

385. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

386. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

387. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

388. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

389. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

390. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

391. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

392. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

393. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

394. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```

395. List Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

396. Set Default Password Aging Policy Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

397. Display Users with No Home Directory:

```
bash awk -F: '$6 == "" {print $1}' /etc/passwd
```

398. Set Maximum Password Age for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS groupname
```

399. Display Users with No Login Shell in /etc/passwd:

```
bash awk -F: '$7 == "/sbin/nologin" {print $1}' /etc/passwd
```

400. Set Default Shell for All Users Non-Interactively:

```
bash sed -i 's/oldshell/newshell/' /etc/passwd
```

401. Show Users in Multiple Groups:

```
bash getent passwd | awk -F: -v groups="group1,group2" '$4 ~ groups {print $1}'
```

402. Set Default Password Aging Policy Non-Interactively:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS -I INACTIVE_DAYS  
username
```

403. Display Users with UID and GID Mismatch:

```
bash awk -F: '$3 != $4 {print $1}' /etc/passwd
```

404. Set Default Password Aging Policy for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS -E EXPIRY_DATE username
```

405. List Users with No Password Aging:

```
bash awk -F: '$8 == "" {print $1}' /etc/shadow
```

406. Set Maximum Inactivity Days for Group:

```
bash chage -I INACTIVE_DAYS groupname
```

407. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

408. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

409. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

410. Set Default Password Aging Policy in Days for All Users:

```
```bash
```

```
chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

411. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

412. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS groupname
```

413. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

414. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

415. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

416. Set Default Shell for All Users Interactively:

bash chsh -s /path/to/shell

417. List Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

418. Set Default Password Aging Policy Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

419. Display Users with No Home Directory:

bash awk -F: '\$6 == "" {print \$1}' /etc/passwd

420. Set Maximum Password Age for Group:

bash chage -M MAX\_DAYS -m MIN\_DAYS groupname

421. Display Users with No Login Shell in /etc/passwd:

bash awk -F: '\$7 == "/sbin/nologin" {print \$1}' /etc/passwd

422. Set Default Shell for All Users Non-Interactively:

bash sed -i 's/oldshell/newshell/' /etc/passwd

423. Show Users in Multiple Groups:

bash getent passwd | awk -F: -v groups="group1,group2" '\$4 ~ groups {print \$1}'

424. Set Default Password Aging Policy Non-Interactively:

bash chage -M MAX\_DAYS -m MIN\_DAYS -W WARN\_DAYS -I INACTIVE\_DAYS  
username

425. Display Users with UID and GID Mismatch:

bash awk -F: '\$3 != \$4 {print \$1}' /etc/passwd

426. Set Default Password Aging Policy for All Users:

bash chage -M MAX\_DAYS -m MIN\_DAYS -I INACTIVE\_DAYS -W WARN\_DAYS -E  
EXPIRY\_DATE username

427. List Users with No Password Aging:

bash awk -F: '\$8 == "" {print \$1}' /etc/shadow

428. Set Maximum Inactivity Days for Group:

bash chage -I INACTIVE\_DAYS groupname

429. Display Users with UID and GID Matching:

bash awk -F: '\$3 == \$4 {print \$1}' /etc/passwd

430. Set Password Expiry Warning Days for Group:

```
bash chage -W WARN_DAYS groupname
```

431. List Users with Specific Shell:

```
bash awk -F: -v shell="/path/to/shell" '$NF == shell {print $1}' /etc/passwd
```

432. Set Default Password Aging Policy in Days for All Users:

```
bash chage -M MAX_DAYS -m MIN_DAYS -W WARN_DAYS username
```

433. Display Users with UID and GID Matching:

```
bash awk -F: '$3 == $4 {print $1}' /etc/passwd
```

434. Set Default Password Aging Policy for Group:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS -W WARN_DAYS  
groupname
```

435. List Users with Expired Accounts:

```
bash awk -F: '{print $1}' /etc/passwd | xargs -l {} chage -l {} | grep "Account expires"
```

436. Set Group Password Aging Policy:

```
bash chage -M MAX_DAYS -m MIN_DAYS -I INACTIVE_DAYS groupname
```

437. Display Users with No Home Directory Ownership:

```
bash find /home -maxdepth 1 -type d -not -user root
```

438. Set Default Shell for All Users Interactively:

```
bash chsh -s /path/to/shell
```