

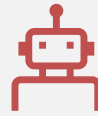


User and Group Management Automation in Kali Linux

By: Adarshvardhan singh

Environment: Kali Linux | Shell Script

Objective



- Automate user and group management



- Configure project-specific directories



- Set permissions and ownership



- Maintain system-level logs for auditing

Project Directory Creation

```
mkdir -p  
/projects/dev_project
```

```
mkdir -p  
/projects/test_project
```

```
mkdir -p  
/projects/admin_project
```

Purpose: Set up isolated
directories for each
group.

dev_project Configuration

```
chown :developers /projects/dev_project
```

```
chmod 770 /projects/dev_project
```

Owned by: developers group

Permissions: 770 (full for owner & group)

test_project Configuration

`chown :developers /projects/test_project`

`chmod 750 /projects/test_project`

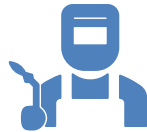
Owned by: developers group

Permissions: 750 (read/execute for others)

admin_project Configuration



chown :admins
/projects/admin_project



chmod 700
/projects/admin_project



Owned by: admins
group



Permissions: 700
(private access)



Shared Configuration File

- `touch /projects/dev_project/config.txt`
- `chmod 660 config.txt`
- Used for shared config
- Permissions: 660 (read/write for owner & group)

User Assignments

```
usermod -aG testers dev1
```

```
usermod -aG admins dev2
```

Added dev1 to testers

Added dev2 to admins

Shared
Directory
with Sticky
Bit

```
mkdir -p  
/projects/shared
```

```
chown :developers  
/projects/shared
```

```
chmod 1770  
/projects/shared
```

Sticky bit ensures only
owner can delete files

Logging & Verification

```
ls -ld /projects/* >> $LOG_FILE
```

```
getent group developers >>  
$LOG_FILE
```

Verifies final permissions and memberships

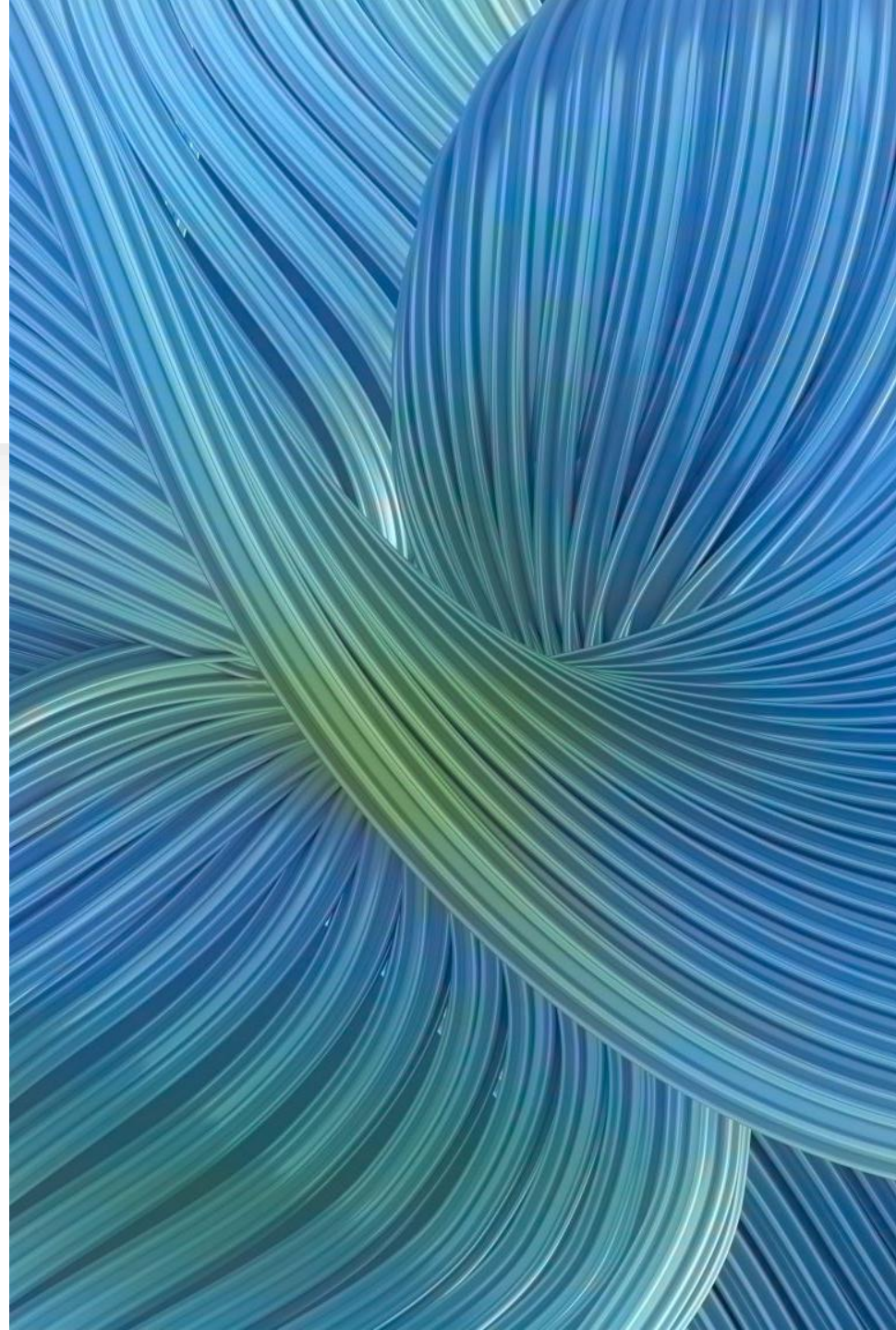
Logs written to
/var/log/user_management.log

Script Execution

```
echo "Script execution completed.  
Check  
/var/log/user_management.log for  
details"
```

Complete automation

Helpful for audits and debugging





root@kali: /home/kali

File Actions Edit View Help

(kali@kali)-[~]

\$ sudo bash

[sudo] password for kali:

(root@kali)-[/home/kali]

nano user_management.sh

File Actions Edit View Help

GNU nano 8.3 user_management.sh

```

echo "Set passwords for all users" >> $LOG_FILE
mkdir -p /projects/dev_project
mkdir -p /projects/test_project
mkdir -p /projects/admin_project
echo "Created project directories" >> $LOG_FILE
chown :developers /projects/dev_project
chmod 770 /projects/dev_project
echo "Set dev_project permissions: 770, owned by developers group" >> $LOG_F

chown :developers /projects/test_project
chmod 750 /projects/test_project
echo "Set test_project permissions: 750, owned by developers group" >> $LOG_

chown :admins /projects/admin_project
chmod 700 /projects/admin_project
echo "Set admin_project permissions: 700, owned by admins group" >> $LOG_FILE

echo "This is a shared configuration file" > /projects/dev_project/config.txt
chown dev1:developers /projects/dev_project/config.txt
chmod 660 /projects/dev_project/config.txt
echo "Created and configured config.txt with 660 permissions" >> $LOG_FILE

usermod -aG testers dev1
usermod -aG admins dev2

```

^G Help
^X Exit

^O Write Out
^R Read File

^F Where Is
^_ Replace

^K Cut
^U Paste

^T Execute
^J Justify

root@kali: /home/kali

File Actions Edit View Help

GNU nano 8.3 user_management.sh

```

chmod 660 /projects/dev_project/config.txt
echo "Created and configured config.txt with 660 permissions" >> $LOG_FILE

```

```

usermod -aG testers dev1
usermod -aG admins dev2
echo "Added dev1 to testers group and dev2 to admins group" >> $LOG_FILE

```

```

mkdir -p /projects/shared
chown :developers /projects/shared
chmod 1770 /projects/shared
echo "Created shared directory with sticky bit (1770)" >> $LOG_FILE

```

```

echo "Final permissions and ownership:" >> $LOG_FILE
ls -ld /projects/* >> $LOG_FILE
ls -l /projects/dev_project/config.txt >> $LOG_FILE

```

```

echo "Group memberships:" >> $LOG_FILE
getent group developers >> $LOG_FILE
getent group testers >> $LOG_FILE
getent group admins >> $LOG_FILE

```

```

echo "User and group management completed successfully - $(date)" >> $LOG_FI
echo "Script execution completed. Check /var/log/user_management.log for det

```

^G Help
^X Exit

^O Write Out
^R Read File

^F Where Is
^_ Replace

^K Cut
^U Paste

^T Execute
^J Justify

GNU nano 8.3 user_management.sh

```

Set -e

LOG_FILE="/var/log/user_management.log"
echo "Starting user and group management - $(date)" > $LOG_FILE

umask 022
echo "Default umask set to 022" >> $LOG_FILE

groupadd developers
groupadd testers
groupadd admins
echo "Created groups: developers, testers, admins" >> $LOG_FILE

useradd -m -g developers -s /bin/bash dev1
useradd -m -g developers -s /bin/bash dev2
useradd -m -g testers -s /bin/bash test1
useradd -m -g admins -s /bin/bash admin1
echo "Created users: dev1, dev2, test1, admin1" >> $LOG_FILE

echo "dev1:devpass123" | chpasswd
echo "dev2:devpass123" | chpasswd
echo "test1:testpass123" | chpasswd
echo "admin1:adminpass123" | chpasswd
echo "Set passwords for all users" >> $LOG_FILE

```

^G Help
^X Exit

^O Write Out
^R Read File

^F Where Is
^_ Replace

^K Cut
^U Paste

^T Execute
^J Justify

[Read 65 lines]

Conclusion

- - Streamlines user/group setup
- - Enforces strict permissions
- - Improves security and efficiency
- - Ideal for multi-user Linux environments

