**HOW TO SETUP A CRON JOB**

In alignment with security compliance standards, the Nautilus project team has opted to impose restrictions on crontab access. Specifically, only designated users will be permitted to create or update cron jobs.

Configure crontab access on App Server 3 as follows: Allow crontab access to john user while denying access to the rod user.

**SOLUTION:**

**Crontab Access Configuration Steps - App Server 3**

**Overview**

Configure crontab access controls to allow only designated users to create or modify cron jobs, enhancing security compliance.

**Prerequisites**

* Root or sudo access on App Server 3
* Knowledge of current user accounts (john and rod)

**Configuration Steps**

**Step 1: Connect to App Server 3**

bash

ssh root@app-server-3

*# Or use your preferred method to access the server*

**Step 2: Navigate to Cron Configuration Directory**

bash

cd /etc

**Step 3: Create/Configure cron.allow File**

The cron.allow file specifies which users are permitted to use crontab.

bash

*# Create or edit the cron.allow file*

sudo nano /etc/cron.allow

Add the following content:

john

Save and exit the editor.

**Step 4: Create/Configure cron.deny File**

The cron.deny file explicitly denies crontab access to specified users.

bash

*# Create the cron.deny file with the rod user*

echo "rod" | sudo tee /etc/cron.deny

*# Alternative method using nano editor:*

*# sudo nano /etc/cron.deny*

*# Add: rod*

*# Save and exit*

**Step 5: Set Proper File Permissions**

Ensure the configuration files have appropriate permissions:

bash

*# Set permissions for cron.allow*

sudo chmod 644 /etc/cron.allow

sudo chown root:root /etc/cron.allow

*# Verify cron.deny was created, then set permissions*

ls -la /etc/cron.deny

*# Set permissions for cron.deny*

sudo chmod 644 /etc/cron.deny

sudo chown root:root /etc/cron.deny

**Step 6: Restart Cron Service**

Restart the cron daemon to apply the new configuration:

bash

*# For systemd-based systems (most modern Linux distributions)*

sudo systemctl restart cron

*# Alternative for some distributions*

sudo systemctl restart crond

*# For older systems using SysV init*

sudo service cron restart

**Step 7: Verification Steps**

**Verify cron.allow configuration:**

bash

cat /etc/cron.allow

*# Should display: john*

**Verify cron.deny configuration:**

bash

cat /etc/cron.deny

*# Should display: rod*

**Test access for john user:**

bash

*# Switch to john user (or login as john)*

sudo su - john

*# Test crontab access*

crontab -l

*# Should work without errors (may show "no crontab for john" if no jobs exist)*

*# Exit back to root/admin user*

exit

**Test access denial for rod user:**

bash

*# Switch to rod user (or login as rod)*

sudo su - rod

*# Test crontab access*

crontab -l

*# Should display error: "You (rod) are not allowed to use this program (crontab)"*

*# Exit back to root/admin user*

exit

**Important Notes**

**Access Control Logic**

1. **If cron.allow exists**: Only users listed in this file can use crontab
2. **If cron.allow doesn't exist but cron.deny does**: All users except those in cron.deny can use crontab
3. **If both files exist**: cron.allow takes precedence
4. **If neither file exists**: Only root can use crontab (most restrictive)

**Security Best Practices**

* Always use cron.allow for explicit permission control
* Keep the user list in cron.allow minimal
* Regularly audit cron access permissions
* Monitor cron job changes through system logs

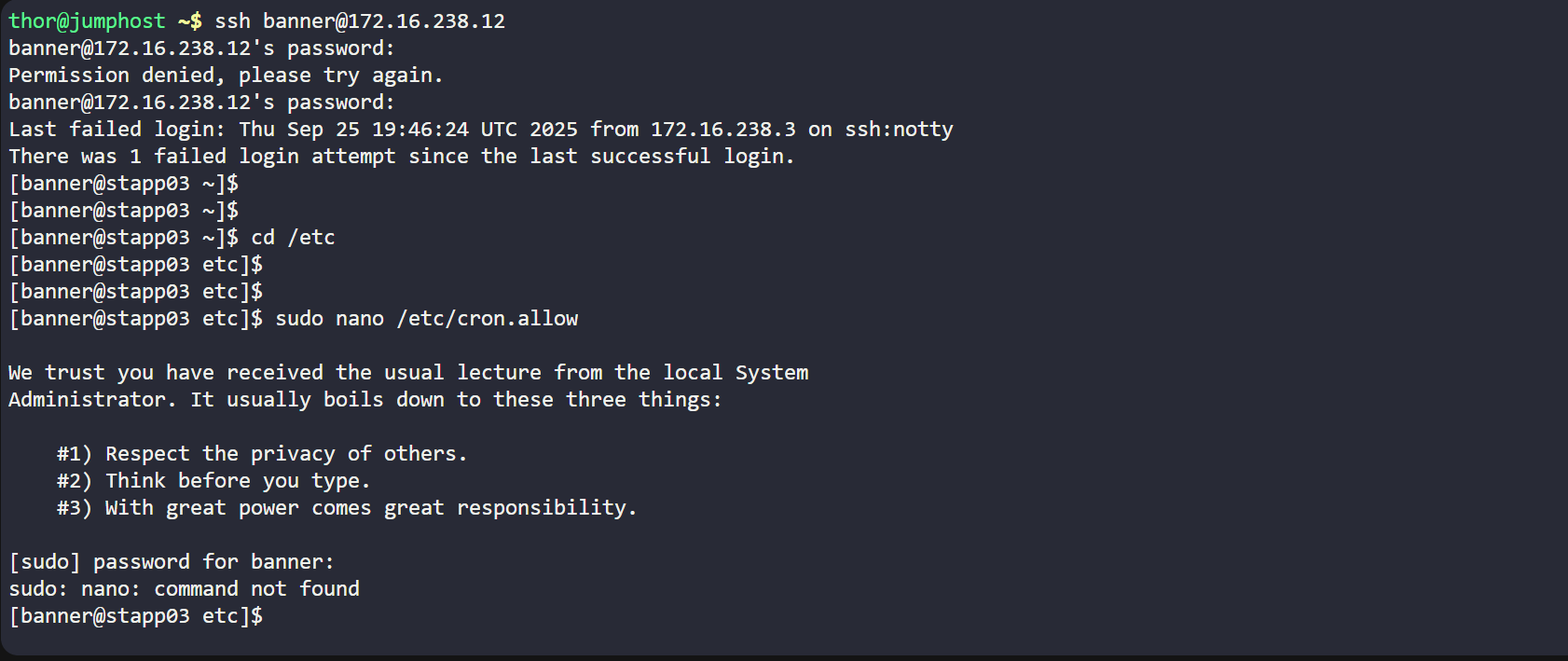
**Troubleshooting**

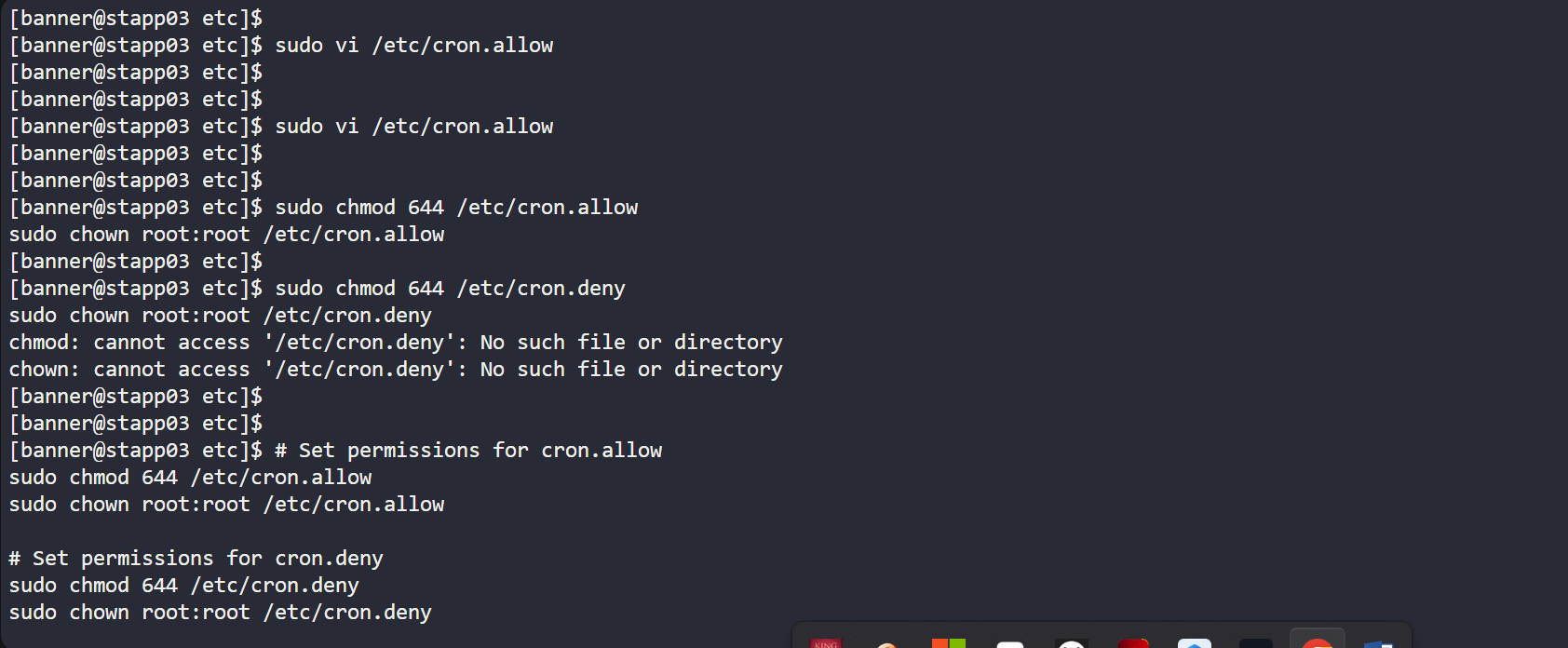
* If cron service fails to restart, check system logs: sudo journalctl -u cron
* Ensure file paths are correct (/etc/cron.allow and /etc/cron.deny)
* Verify user names are spelled correctly and exist on the system

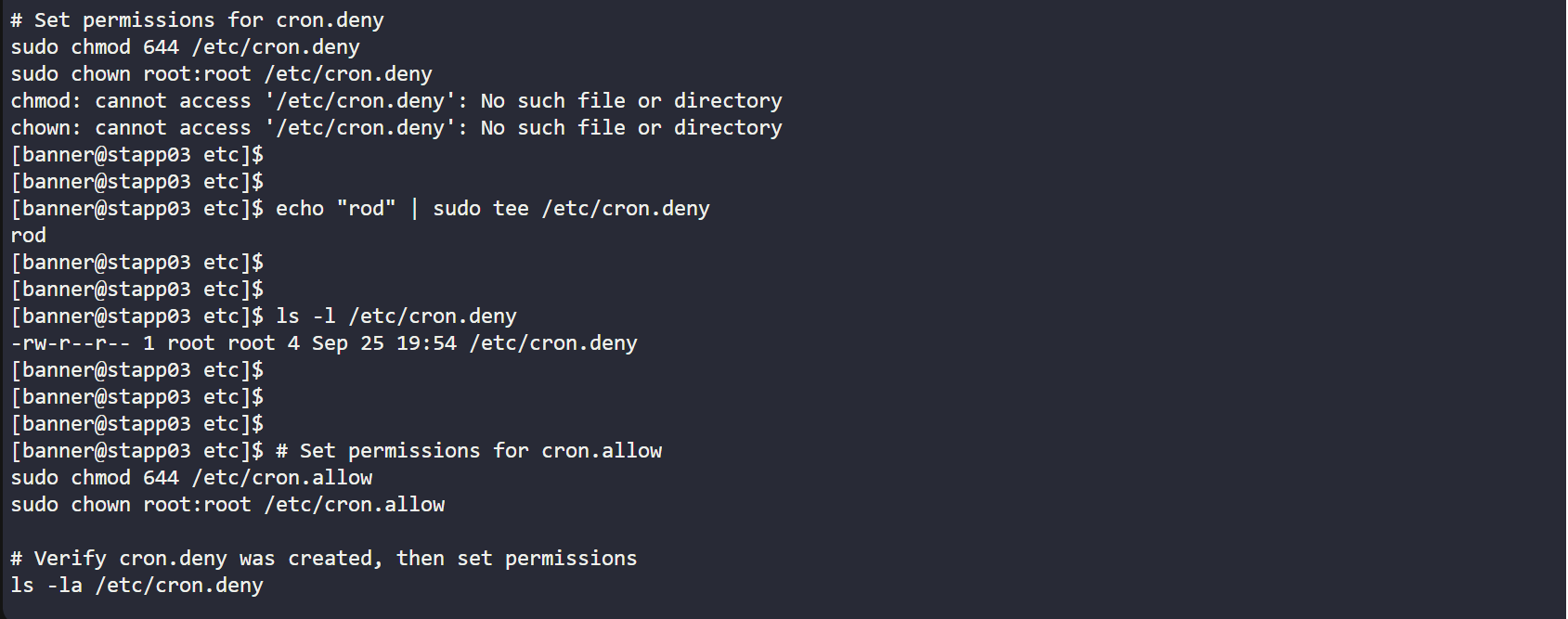
**Verification Complete**

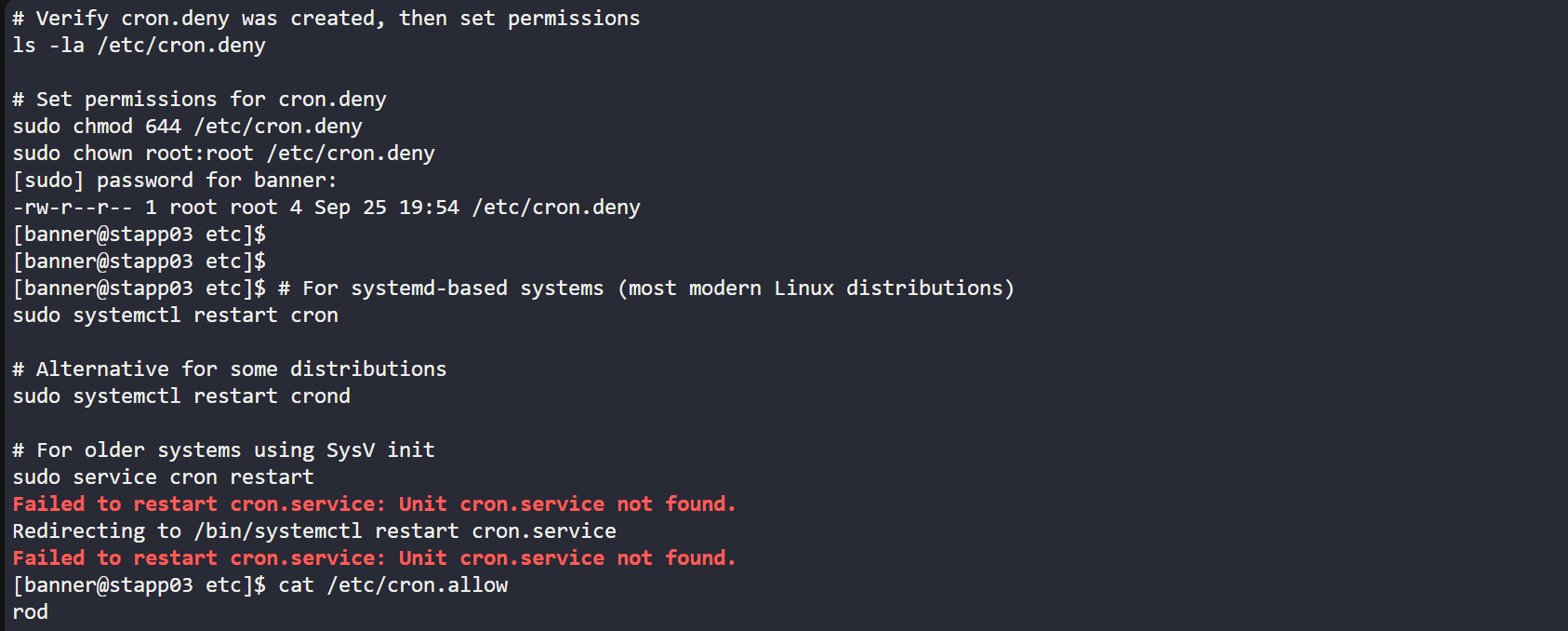
After completing these steps:

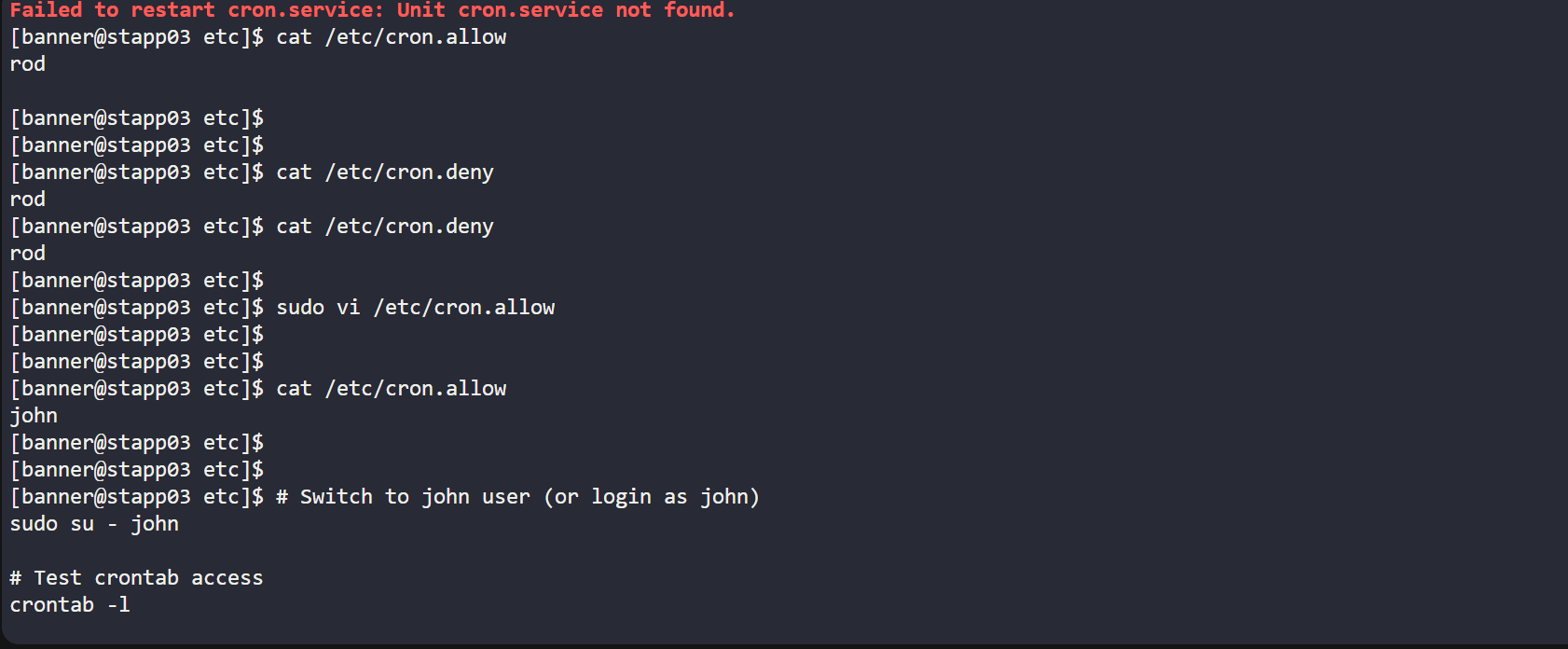
* ✅ john user can create and modify cron jobs
* ❌ rod user is denied crontab access
* 🔒 Security compliance requirements are met

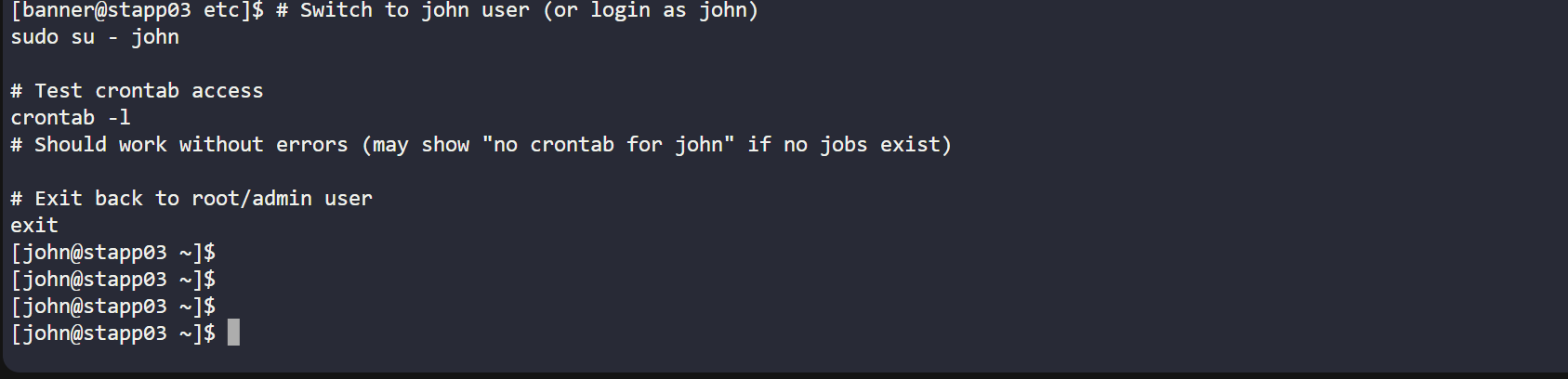




****

****

****

****