Incident Report

1. When the Cron Job Was Added

- The cron job was manually added to **/var/spool/cron/root**, bypassing the standard **crontab -e** command.
- No exact timestamp was logged in this simulation, but in real-world scenarios, this can be verified using system logs such as **/var/log/syslog** or **/var/log/cron**.

2. What the Script Was Doing

- The script (**/tmp/malicious.sh**) wrote a message to a hidden log file every minute:
    - **echo "Ping from attacker server" >> /tmp/.cron.log**
- This mimics attacker behaviour for maintaining persistence and silently logging activity.

3. Any Signs of Lateral Movement or Download Activity

- In this simulation, no outbound connections, lateral movement, or file downloads were observed.
- In a real environment, check with tools and logs such as:
    - **lsof, netstat, ss** (for suspicious connections)
    - **.bash_history, /var/log/auth.log** (for user activity)
    - **/var/log/syslog** or endpoint EDR logs for download traces.

Recommendations

1. **Restrict Cron Job Access**
    - Limit cron job editing rights to only system administrators.
    - Secure cron directories and files with strict file permissions.
2. **Enable Cron Integrity Checks**

- Use tools like **AIDE** or **Tripwire** to monitor changes in cron directories and system binaries.

3. **Set Up Alerts for New Cron Entries**
    - Implement **auditd** rules or **inotify** watches to detect and alert on:
        - New or modified cron jobs
        - Unauthorized script executions from **/tmp** or similar paths