Task Scheduling in Linux

Overview

Task scheduling is a critical feature in Linux systems that allows users and administrators to automate tasks by running them at specific times or regular intervals, eliminating the need for manual initiation.

Common uses include:

- Automatic software updates
- Script execution
- Database maintenance
- Backup automation

It ensures consistent execution of tasks and can notify users of events.

> Task scheduling is like setting a coffee maker to brew automatically each morning.

Understanding task scheduling is essential for cybersecurity specialists to:

- Detect unauthorized cron jobs or malicious persistence
- Enhance system audits
- Simulate attack scenarios

Systemd

Systemd is a system and service manager for Linux systems like Ubuntu and Red Hat. It can automate tasks using **timers** and **services**.

Steps to schedule with systemd

1. **Create a Timer**

```
sudo mkdir /etc/systemd/system/mytimer.timer.d
sudo vim /etc/systemd/system/mytimer.timer
```

mytimer.timer

```
[Unit]
Description=My Timer

[Timer]
OnBootSec=3min
OnUnitActiveSec=1hour
```

```
[Install]
WantedBy=timers.target
```

1. **Create a Service**

```
sudo vim /etc/systemd/system/mytimer.service
```

mytimer.service

```
[Unit]
Description=My Service

[Service]
ExecStart=/full/path/to/my/script.sh

[Install]
WantedBy=multi-user.target
```

1. **Reload systemd and start**

```
sudo systemctl daemon-reload
sudo systemctl start mytimer.timer
sudo systemctl enable mytimer.timer
```

Cron

Cron is a daemon that executes scheduled commands from a crontab file.

Crontab Time Format

Field	Value	Description
Minute	0-59	Minute of the hour
Hour	0-23	Hour of the day
Day of Month	1-31	Day of the month
Month	1-12	Month of the year
Day of Week	0-7 (Sun=0/7)	Day of the week

Example Crontab Entries

```
# System Update every 6 hours
0 */6 * * * /path/to/update_software.sh

# Execute scripts on the 1st of each month at midnight
0 0 1 * * /path/to/scripts/run_scripts.sh

# Cleanup database every Sunday at midnight
0 0 * * 0 /path/to/scripts/clean_database.sh

# Perform backups every Sunday at midnight
0 0 * * 7 /path/to/scripts/backup.sh
```

Notifications and Logs

- You can configure email notifications for cron job results.
- Logging can be enabled to monitor task execution status and history.

Systemd vs Cron

Feature	Systemd	Cron
Configuration	Uses `.service` and `.timer` files	Uses `crontab` entries
Flexibility	More complex, supports triggers	Simpler, time-based only
Logging	Integrated journal logging	Requires external log setup
Usage	Preferred for new systems	Still widely used and simpler