Avvio del sistema

In questo capitolo imparerai come si avvia il sistema.

Obiettivi : In questo capitolo, i futuri amministratori Linux apprenderanno:

✔ Le diverse fasi del processo di avvio;
✔ Come Rocky Linux supporta questo avvio tramite Grub2 e systemd;
✔ Come proteggere Grub2 da un attacco;
✔ Come gestire i servizi;
✔ Come accedere ai registri di log con journald.

🏁 **utenti**

Conoscenza: ⭐⭐
Complessità: ⭐⭐⭐

Tempo di lettura: 20 minuti

Il processo di avvio

È importante capire il processo di avvio di Linux per poter risolvere i problemi che potrebbero verificarsi.

Il processo di avvio include:

L'avvio del BIOS

Il **BIOS** (Basic Input/Output System) esegue il **POST** (power on self test) per rilevare, testare e inizializzare i componenti hardware del sistema.

Quindi carica il **MBR** (Master Boot Record).

Il Master boot record (MBR)

Il Master Boot Record sono i primi 512 byte del disco di avvio. Il MBR trova il dispositivo di avvio e carica il bootloader **GRUB2** in memoria passando il controllo ad esso.

I successivi 64 byte contengono la tabella delle partizioni del disco.

Il bootloader Grub2

Il bootloader predefinito per la distribuzione Rocky 8 è **GRUB2** (GRand Unified Bootloader). GRUB2 sostituisce il vecchio. GRUB bootloader (chiamato anche GRUB legacy).

Il file di configurazione di GRUB2 si trova in `/boot/grub2/grub.cfg` ma questo file non dovrebbe mai essere modificato direttamente.

Le impostazioni di configurazione del menu GRUB2 si trovano in `/etc/default/grub` e sono usate per generare il file `grub.cfg`.

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root
rhgb quiet net.ifnames=0"
GRUB_DISABLE_RECOVERY="true"
```

Se vengono apportate modifiche a uno o più di questi parametri, deve essere eseguito il comando `grub2-mkconfig` per rigenerare il file `/boot/grub2/grub.cfg`.

```
[root] # grub2-mkconfig -o /boot/grub2/grub.cfg
```

- GRUB2 cerca l'immagine del kernel compresso (il file `vmlinuz`) nella cartella `/boot`.
- GRUB2 carica l'immagine del kernel in memoria ed estrae il contenuto del file immagine `initramfs` in una cartella temporanea in memoria usando il file system `tmpfs`.

Il kernel

Il kernel inizia il processo `systemd` con PID 1.

```
root          1      0  0 02:10 ?        00:00:02 /usr/lib/systemd/systemd
--switched-root --system --deserialize 23
```

`systemd`

Systemd è il genitore di tutti i processi di sistema. Legge il target del link `/etc/systemd/system/default.target` (es. `/usr/lib/systemd/system/multi-user.target`) per determinare l'obiettivo predefinito del sistema. Il file definisce i servizi da avviare.

Systemd posiziona quindi il sistema nello stato definito dall'obiettivo eseguendo le seguenti attività di inizializzazione:

1. Imposta il nome della macchina

2. Inizializza la rete
3. Inizializza SELinux
4. Mostra il banner di benvenuto
5. Inizializza l'hardware in base agli argomenti forniti al kernel al momento dell'avvio
6. Monta i file system, inclusi i file system virtuali come /proc
7. Pulisce le directory in /var
8. Avvia la memoria virtuale (swap)

Protezione del bootloader GRUB2

Perché proteggere il bootloader con una password?

1. Prevenire l'accesso in ***Single user mode*** - Se un utente malintenzionato può avviare in single user mode, diventa l'utente root.
2. Impedire l'accesso alla console di GRUB - Se un utente malintenzionato riesce a utilizzare la console Grub, può modificare la sua configurazione o raccogliere informazioni sul sistema utilizzando il comando `cat`.
3. Impedire l'accesso ai sistemi operativi insicuri. Se c'è un doppio avvio sul sistema, un utente malintenzionato può selezionare un sistema operativo come DOS che all'avvio ignora i controlli di accesso e le autorizzazioni dei file.

Per proteggere con password il bootloader GRUB2:

- Rimuovere `-unrestricted` dalla dichiarazione principale `CLASS=` nel file `/etc/grub.d/10_linux`.
- Se un utente non è stato ancora configurato, utilizzare il comando `grub2-setpassword` per fornire una password per l'utente root:

```
# grub2-setpassword
```

Un file `/boot/grub2/user.cfg` sarà creato se non era già presente. Contiene la password hashed. di GRUB2.

> **Nota**
>
> Questo comando supporta solo le configurazioni con un singolo utente root.

```
[root]# cat /boot/grub2/user.cfg
```

```
GRUB2_PASSWORD=grub.pbkdf2.sha512.10000.CC6F56....A21
```

- Ricreare il file di configurazione con il comando `grub2-mkconfig`:

```
[root]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-327.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-327.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-f9725b0c842348ce9e0bc81968cf7181
Found initrd image: /boot/initramfs-0-rescue-
f9725b0c842348ce9e0bc81968cf7181.img
done
```

- Riavviare il sistema e controllare.

Tutte le voci definite nel menu GRUB richiederanno ora un utente e una password da inserire a ciascun avvio. Il sistema non avvierà un kernel senza l'intervento diretto dell'utente dalla console.

- Quando viene richiesto l'utente, inserire `root`;
- Quando viene richiesta una password, inserire la password fornita al comando `grub2-setpassword`.

Per proteggere solo la modifica delle voci del menu GRUB e l'accesso alla console, l'esecuzione del comando `grub2-setpassword` è sufficiente. Ci possono però essere casi in cui ci sono buone ragioni per non farlo. Questo potrebbe essere particolarmente vero in un data center remoto in cui l'inserimento di una password ogni volta che viene riavviato un server è difficile o impossibile da fare.

Systemd

Systemd è un gestore di servizi per i sistemi operativi Linux.

È sviluppato per:

- rimanere compatibile con gli script di inizializzazione del vecchio SysV,
- fornire molte funzionalità, come l'avvio parallelo dei servizi di sistema all'avvio del sistema, l'attivazione su richiesta dei demoni, il supporto per le istantanee o la gestione delle dipendenze tra i servizi.

> **Nota**
>
> Systemd è il sistema di inizializzazione predefinito da RedHat/CentOS 7.

Systemd introduce il concetto di unità systemd.

Tipo	Estensione del file	Osservazioni
Unità di servizio	`.service`	Servizio di sistema
Unità di destinazione	`.target`	Un gruppo di unità systemd
Mount unit	`.automount`	Un punto di montaggio automatico per il file system

> **Nota**
>
> Ci sono molti tipi di unità: Device unit, Mount unit, Path unit, Scope unit, Slice unit, Snapshot unit, Socket unit, Swap unit, Timer unit.

- Systemd supporta le istantanee dello stato del sistema e il ripristino.
- Mount points possono essere configurati come target di systemd.
- All'avvio, systemd crea socket di ascolto per tutti i servizi di sistema che supportano questo tipo di attivazione e passa questi socket ai relativi servizi non appena vengono avviati. Ciò consente di riavviare un servizio senza perdere un singolo messaggio inviato dalla rete durante la sua indisponibilità. Il socket corrispondente rimane accessibile e tutti i messaggi vengono accodati.
- I servizi di sistema che utilizzano D-BUS per le comunicazioni tra processi possono essere avviati su richiesta la prima volta che vengono utilizzati da un client.
- Systemd arresta o riavvia solo i servizi in esecuzione. Le versioni precedenti (prima di RHEL7) tentavano di arrestare direttamente i servizi senza controllarne lo stato corrente.
- I servizi di sistema non ereditano alcun contesto (come le variabili di ambiente HOME e PATH). Ogni servizio opera nel proprio contesto di esecuzione.

Tutte le operazioni delle unità di servizio sono soggette a un timeout predefinito di 5 minuti per evitare che un servizio malfunzionante blocchi il sistema.

Gestione dei servizi di sistema

Le unità di servizio terminano con l'estensione di file `.service` e hanno uno scopo simile a

quello degli script di init. Il comando `systemctl` viene utilizzato per `visualizzare`, `avviare`, `fermare`, `riavviare` un servizio di sistema:

systemctl	**Descrizione**
systemctl start ***name***.service	Avvia un servizio
systemctl stop ***name***.service	Arresta un servizio
systemctl restart ***name***.service	Riavvia un servizio
systemctl reload ***name***.service	Ricarica una configurazione
systemctl status ***name***.service	Controlla se un servizio è in esecuzione
systemctl try-restart ***name***.service	Riavvia un servizio solo se è in esecuzione
systemctl list-units --type service --all	Visualizza lo stato di tutti i servizi

Il comando `systemctl` viene utilizzato anche per `abilitare` o `disabilitare` un servizio di sistema e la visualizzazione dei servizi associati:

systemctl	**Descrizione**
systemctl enable ***name***.service	Attivare un servizio
systemctl disable ***name***.service	Disabilitare un servizio
systemctl list-unit-files --type service	Elenca tutti i servizi e i controlli se sono in esecuzione
systemctl list-dependencies --after	Elenca i servizi che si avviano prima dell'unità specificata
systemctl list-dependencies --before	Elenca i servizi che si avviano dopo l'unità specificata

Esempi:

```
systemctl stop nfs-server.service
# or
```

```
systemctl stop nfs-server
```

Per elencare tutte le unità attualmente caricate:

```
systemctl list-units --type service
```

Per elencare tutte le unità e per verificare se sono attivate:

```
systemctl list-unit-files --type service
```

```
systemctl enable httpd.service
systemctl disable bluetooth.service
```

Esempio di un file .service per il servizio postfix

```
postfix.service Unit File
What follows is the content of the /usr/lib/systemd/system/postfix.service
unit file as currently provided by the postfix package:

[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop

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

Utilizzo degli obiettivi di sistema

Su Rocky8/RHEL8, il concetto di runlevel è stato sostituito dagli obiettivi systemd.

I sistemi di destinazione sono rappresentati da unità di destinazione. Le unità di destinazione terminano con l'estenzione `.target` e il loro unico scopo è di raggruppare altre unità systemd in una catena di dipendenze.

Ad esempio, l'unità `graphical.target`, che viene utilizzata per avviare una sessione grafica, inizializza i servizi di sistema come il **GNOME display manager** (`gdm.service`) o l'**accounts service** (`accounts-daemon.service`) e attiva anche l'unità `multi-user.target` .

Allo stesso modo, l'unità `multi-user.target` inizializza altri servizi di sistema essenziali, come **NetworkManager** (`NetworkManager.service`) o **D-Bus** (`dbus.service`) e attiva un'altra unità di destinazione denominata `basic.target` .

Unità di destinazione.	**Descrizione**
poweroff.target	Chiude il sistema e lo spegne
rescue.target	Attiva una shell di salvataggio
multi-user.target	Attiva un sistema multiutente senza interfaccia grafica
graphical.target	Attiva un sistema multiutente con interfaccia grafica
reboot.target	Spegne e riavvia il sistema

La destinazione predefinita

Per determinare quale obiettivo viene utilizzato per impostazione predefinita:

```
systemctl get-default
```

Questo comando cerca l'obiettivo del collegamento simbolico situato in `/etc/systemd/system/default.target` e visualizza il risultato.

```
$ systemctl get-default
graphical.target
```

Il comando `systemctl` può anche fornire un elenco di obiettivi disponibili:

```
systemctl list-units --type target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
bluetooth.target       loaded active active Bluetooth
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target  loaded active active Network is Online
network.target         loaded active active Network
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
```

```
sound.target           loaded active active Sound Card
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers
```

Per configurare il sistema all'utilizzo di un diverso target predefinito:

```
systemctl set-default name.target
```

Esempio:

```
# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system
/default.target'
```

Per passare a un'unità di destinazione diversa nella sessione corrente:

```
systemctl isolate name.target
```

La **Modalità di ripristino** fornisce un ambiente semplice per riparare il sistema nei casi in cui è impossibile eseguire un normale processo di avvio.

In `modalità di ripristino`, il sistema tenta di montare tutti i file system locali e avviare diversi servizi di sistema importanti, ma non abilita un'interfaccia di rete o consente ad altri utenti di connettersi al sistema contemporaneamente.

Su Rocky 8, la `modalità di ripristino` è equivalente al vecchio `single user mode` e richiede la password di root.

Per modificare la destinazione corrente immettere `rescue mode` nella sessione corrente:

```
systemctl rescue
```

Modalità di emergenza fornisce l'ambiente più minimalista possibile e consente di riparare il sistema anche in situazioni in cui il sistema non è in grado di inserire la modalità di salvataggio. Nella modalità di emergenza, il sistema monta il file system root solo per la lettura. Non tenterà di montare qualsiasi altro file system locale, non attiverà alcuna interfaccia di rete e inizializzerà alcuni servizi essenziali.

Per modificare il target corrente e immettere la modalità di emergenza nella sessione corrente:

```
systemctl emergency
```

Arresto, sospensione e ibernazione

Il comando `systemctl` sostituisce alcuni dei comandi di gestione dell'alimentazione utilizzati

nelle versioni precedenti:

Vecchio comando	Nuovo comando	Descrizione
`halt`	`systemctl halt`	Spegne il sistema.
`poweroff`	`systemctl poweroff`	Arresta elettricamente il sistema.
`reboot`	`systemctl reboot`	Riavvia il sistema.
`pm-suspend`	`systemctl suspend`	Sospende il sistema.
`pm-hibernate`	`systemctl hibernate`	Iberna il sistema.
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Iberna e sospende il sistema.

Il processo `journald`

I file di registro possono, oltre a `rsyslogd`, essere gestiti anche dal demone `journald` che è un componente di `systemd`.

Il demone `journald` cattura i messaggi Syslog, i messaggi di registro del kernel, i messaggi dal disco RAM iniziale e dall'inizio dell'avvio, nonché i messaggi scritti nell'output standard e l'output di errore standard di tutti i servizi, quindi li indicizza e li rende disponibili all'utente.

Il formato del file di registro nativo, che è un file binario strutturato e indicizzato, migliora le ricerche e consente un funzionamento più rapido, memorizza anche le informazioni dei metadati, come i timestamp o gli ID utente.

comando `journalctl`

Il comando `journalctl` visualizza i file di registro.

```
journalctl
```

Il comando elenca tutti i file di registro generati sul sistema. La struttura di questa uscita è simile a quella utilizzata in `/var/log/messages/` ma offre alcuni miglioramenti:

- la priorità delle voci è segnata visivamente;
- i timestamp sono convertiti nella zona oraria locale del sistema;
- vengono visualizzati tutti i dati registrati, inclusi i registri rotativi;

- l'inizio di un avvio è contrassegnato da una linea speciale.

Uso del display continuo

Con il display continuo, i messaggi di registro vengono visualizzati in tempo reale.

```
journalctl -f
```

Questo comando restituisce un elenco delle dieci linee di registro più recenti. L'utilità continua quindi a funzionare e attende che avvengano nuove modifiche per visualizzarle immediatamente.

Filtrare i Messaggi

È possibile utilizzare diversi metodi di filtraggio per estrarre informazioni che si adattano a diverse esigenze. I messaggi di registro vengono spesso utilizzati per monitorare il comportamento errato del sistema. Per visualizzare le voci con una priorità selezionata o superiore:

```
journalctl -p priority
```

È necessario sostituire la priorità con una delle seguenti parole chiave (o un numero):

- debug (7),
- info (6),
- notice (5),
- warning (4),
- err (3),
- crit (2),
- alert (1),
- and emerg (0).

Ultimo aggiornamento: 6 gennaio 2022