Opgave 12 - Linux log system

Information

Formålet med denne øvelse er at introducere Linux-logsystemet og præsentere de grundlæggende koncepter. Hver Linux-distribution har sine egne variationer i logsystemet, men grundlæggende fungerer de ens. I dette fag arbejdes der med logsystemet på en Ubuntu-server.

I Linux findes der overordnet to logsystemer: **rsyslog** (som er en nyere version af **syslog**) og **journalctl**. Begge systemer bruges til at håndtere logfiler, men der er forskelle:

- **rsyslog** skriver logs til filer i `/var/log/`.
- **journalctl** håndterer logs via systemd's journal og kan bruges til at søge i logs uden at åbne filer direkte.

I denne opgave fokuserer vi på **rsyslog** og logfilerne i Ubuntu (baseret på Debian).

Logfiler i Ubuntu

Systemets logfiler opbevares i mappen `/var/log/`. To af de vigtigste logfiler er:

- **syslog** – Den primære logfil, der indeholder systemets aktiviteter.
- **auth.log** – Log over autentificering og sikkerhedshændelser.

rsyslog vs. journalctl

Både **rsyslog** og **journalctl** bruges til loghåndtering i Linux, men de har forskellige tilgange og formål:

rsyslog

- En ældre, filbaseret logningsmekanisme, der skriver logs til filer i **/var/log/** (f.eks. `syslog`, `auth.log` osv.).
- Bruger en konfigurationsfil (`/etc/rsyslog.conf`) til at bestemme, hvordan logs behandles og sendes videre (fx til eksterne servere).
- **Fordele**:
- Let at filtrere og analysere med standard Unix-værktøjer (`grep`, `awk`, `sed`).

- Kan sende logs til eksterne systemer via netværksprotokoller.
- Lavt ressourcetræk.

journalctl (systemd-journal)

- En nyere, binær logmekanisme, der er integreret i **systemd**.
- Logs gemmes i et binært format og kan kun læses med `journalctl`.
- Tillader mere avanceret søgning og filtrering (fx logs for en specifik service eller tid).
- **Fordele**:
- Bedre filtrerings- og søgemuligheder (`journalctl -u ssh` for kun at se SSH-relaterede logs).
- Metadata såsom PID, UID og cgroup gemmes automatisk.
- Understøtter persistente og volatile logs.

Hvornår bruger man hvad?

- **rsyslog** er bedst, hvis du arbejder med traditionelle logfiler eller sender logs til eksterne systemer. Fordelen ved rsyslog er, at den skriver loglinjerne som almindelig tekst direkte til logfilerne, hvilket gør det nemmere at integrere logfiler fra rsyslog med andre systemer (såsom SIEM, IDS, log-aggregeringsværktøjer osv.).
- **journalctl** er praktisk til systemd-styrede systemer, da det giver mere avancerede søgefunktioner og bedre integration med services.

Mange moderne Linux-systemer bruger **begge**, hvor journalctl fanger alle logs, mens rsyslog sender dem videre til filbaserede logs.

Instruktioner

1. Primær logfil: `syslog`

Den primære logfil for systemet hedder `syslog` og indeholder information om stort set alt, hvad systemet foretager sig.

1. Udskriv de seneste 20 linjer af `syslog`:

   ```
   tail -n 20 /var/log/syslog
   ```

2. Studér logformatet. En typisk linje i `syslog` ser sådan ud:

   ```
   Mar 12 10:15:03 hostname process[1234]: This is a log message
   ```

   Identificér følgende elementer:

- **Tidsstempel** (fx `Mar 12 10:15:03`)
- **Værtsnavn** (fx `hostname`)
- **Program/Service** (fx `process[1234]`)
- **Besked** (fx `This is a log message`)

3. Udfra de identificeret elementer, lav et generiske log format til dit Linux cheat sheet.
4. Find ud af, hvilken tidszone loggen bruger til tidsstempling:

```
timedatectl
```

5. Er systemets tidszone **UTC**, eller er den lokal?
6. Alternativt kan du tjekke tidsstempler direkte i syslog ved at søge efter "time" med følgende kommando:

```
grep -i "time" /var/log/syslog | head -n 5
```

2. Authentication log: `auth.log`

Logfilen `auth.log` indeholder information om autentificering og sikkerhedshændelser.

1. Udskriv de seneste 20 linjer af `auth.log`:

```
tail -n 20 /var/log/auth.log
```

2. Identificér brugerens navn (fx `root` eller en anden konto).
3. Skift til root og udskriv `auth.log` igen:

```
sudo su
tail -n 20 /var/log/auth.log
```

4. Bemærk, at `sudo su` muligvis ikke altid efterlader en logpost. Prøv også:

```
sudo -i
exit
tail -n 20 /var/log/auth.log
```

5. Skift tilbage til din primære bruger og udskriv `auth.log` igen.
6. Hvilke nye rækker blev tilføjet?
7. Kan du se spor af dit autentificerings-forsøg?

3. Filtrering af logs

I stedet for at udskrive hele logfiler kan du filtrere relevante oplysninger:

- Søg efter fejlmeddelelser i `syslog`:

```
grep -i "error" /var/log/syslog | tail -n 10
```

- Find autentificeringsfejl i `auth.log`:

```
grep -i "failed" /var/log/auth.log | tail -n 10
```

4. Bonusopgave: Find system-genstartere

1. Søg efter alle system-genstartere i `syslog`:

```
grep -i "reboot" /var/log/syslog
```

2. Hvilke tidspunkter genstartede systemet?
3. Kan du identificere brugeren, der udløste genstarten, baseret på logdata? ved at samholde data fra syslog med authlog `grep -i "sudo" /var/log/auth.log | grep -E "reboot|shutdown"`
4. Yderlige kan man også se systemets sidste genstart med kommandoen `last reboot`

5. Brug af journalctl

Ud over `rsyslog` kan du også bruge `journalctl` til at vise logs: - Udskriv de seneste 20 logs fra systemets journal:

```
journalctl -n 20
```

- Filtrér logs for en specifik service, fx SSH:

```
journalctl -u ssh --no-pager | tail -n 10
```

Refleksion

- Hvad kan du lære om systemets aktivitet ud fra disse logs?
- Hvorfor er logfiler vigtige for fejlfinding og sikkerhed?

Links

- rsyslog Documentation

- journalctl Guide

2025-04-03 05:55:59