108.2 & 107.2: Logs do sistema & Temporizadores do systemd

Rafael Obelheiro

06/11/2024

108.2 Logs do sistema

Logs

- Logs são mensagens que registram várias atividades do sistema
  - incluindo o kernel e diferentes serviços
- São muito importantes para detecção e resolução de problemas
- O gerenciamento de logs abrange
  - coletar logs de diversas fontes
  - fornecer uma interface estruturada para consulta, análise, filtragem e monitoramento de mensagens
  - gerenciar a retenção e expiração de mensagens de modo que os dados sejam armazenados enquanto forem potencialmente úteis ou requeridos por lei, mas não além disso

Serviços usados para coleta de logs

- `syslog`: serviço padrão para gerenciamento de logs no Unix
  - logs armazenados em arquivos texto
    - ★ flexibilidade para consulta
  - oferece suporte a loghosts UDP
- `rsyslog`: reimplementação do `syslog`, com vários recursos adicionais
  - mais poderoso e flexível
  - oferece suporte a loghosts UDP e TCP
- `systemd-journald`
  - logs armazenados em formato binário comprimido
    - ★ precisam ser consultados usando ferramentas específicas
  - geralmente usado em conjunto com `syslog`/`rsyslog`

Arquitetura de logging

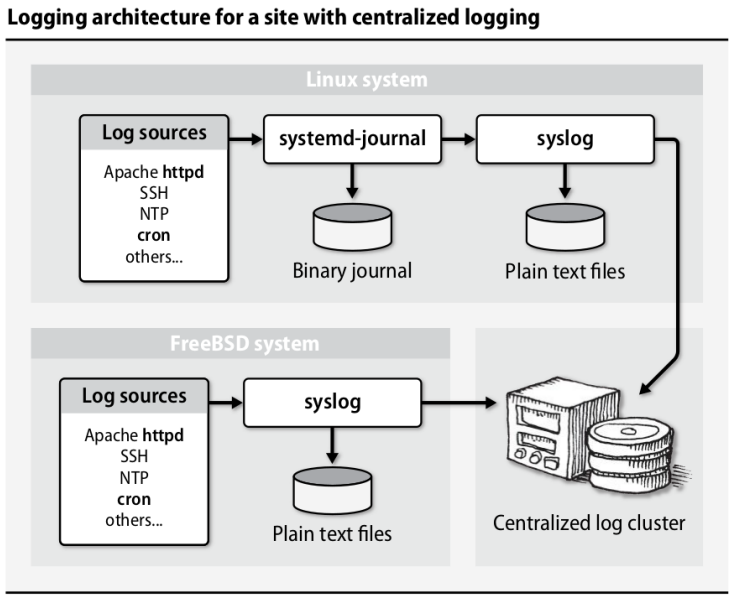

Figure 1:

Logs binários relativos a logins

- `/var/run/utmp` e `/var/log/wtmp`: logins bem sucedidos, mudanças do sistema (shutdowns, reboots, runlevels)
  - cmd: `w`, `who`, `last`
- `/var/log/btmp`, `/var/log/faillog`: logins malsucedidos
  - cmd: `lastb`, `faillog`
    - ★ uso de `/var/log/faillog` desabilitado por padrão no Ubuntu
- `/var/log/lastlog`: data/hora de logins recentes
  - cmd: `lastlog`

Exemplos

```
### Mostra logins e mudanças do sistema/apenas sistema
$ last
$ last -x

### Mostra logins do usuário linus
$ last linus

### Mostra logins malsucedidos (todos/apenas de linus)
$ sudo lastb
$ sudo lastb linus

### Mostra último login de linus
$ lastlog -u linus

### Mostra quem (não) fez login nos últimos 2 dias
$ lastlog -t 2
$ lastlog -b 2
```


Principais logs em `/var/log` (Ubuntu)

- `apache2/*`: logs do Apache
- `apt/*`: logs do APT
- `auth.log`: logins e outros eventos de autenticação de usuários
- `dmesg`: mensagens do kernel no boot atual
  - útil caso `dmesg(1)` não mostre todas as mensagens
- `dpkg.log`: logs do `dpkg`
- `kern.log`: mensagens do kernel
- `mail.*`: serviço de email
- `syslog`: principais logs do sistema
- `Xorg.0.log`: servidor gráfico (X)
- Gerenciados pelo `rsyslog`, em geral
  - alguns serviços escrevem diretamente nos arquivos de log

Configuração do `rsyslog`

- `/etc/rsyslog.conf`: configurações de módulos e diretivas globais do serviço
  - módulos definem como mensagens podem ser recebidas (entrada) e armazenadas (saída)
  - diretivas globais definem atributos gerais e valores default para parâmetros
- `/etc/rsyslog.d/*.conf`: regras de processamento das mensagens de log

Regras de processamento

- Cada mensagem de log tem um recurso (*facility*) e uma prioridade (*priority*) associados
  - o recurso é associado ao subsistema que gerou a mensagem
  - a prioridade indica a importância da mensagem
- As regras de processamento definem ações para as mensagens de log conforme seu recurso e prioridade
- As regras têm o formato

```
recurso.prioridade: ação
```

  - múltiplos recursos podem ser separados por vírgula
    - ★ `rec1,rec2.pri: ação`
  - múltiplos pares podem ser separados por ponto e vírgula
    - ★ `rec1.pri1;rec2.pri2: ação`

Recursos (*facilities*)

- Os recursos podem usar palavras-chave ou números

Syslog facility names

Facility	Programs that use it
*	All facilities except "mark"
auth	Security- and authorization-related commands
authpriv	Sensitive/private authorization messages
cron	The **cron** daemon
daemon	System daemons
ftp	The FTP daemon, **ftpd** (obsolete)
kern	The kernel
local0-7	Eight flavors of local message
lpr	The line printer spooling system
mail	**sendmail, postfix**, and other mail-related software
mark	Time stamps generated at regular intervals
news	The Usenet news system (obsolete)
syslog	**syslogd** internal messages
user	User processes (the default if not specified)

Prioridades (*priority* ou *severity*)

- As prioridades têm uma ordem definida

Syslog severity levels (descending severity)

Level	Approximate meaning
emerg	Panic situations; system is unusable
alert	Urgent situations; immediate action required
crit	Critical conditions
err	Other error conditions
warning	Warning messages
notice	Things that might merit investigation
info	Informational messages
debug	For debugging only

Figure 3:

Especificação de prioridades

- Quando a prioridade não é precedida por um modificador, a regra casa com mensagens de prioridade **igual ou superior** à especificada
  - `crit`: `crit` + `alert` + `emerg`
  - `info`: qualquer prioridade exceto `debug`
- `none` exclui um recurso
  - `*.err;kernel.none`: mensagens que não sejam do recurso `kernel` e com prioridade $\geq$ `err`

Examples of priority level qualifiers

Selector	Meaning
`auth.info`	Auth-related messages of info priority and higher
`auth.=info`	Only messages at info priority
`auth.info;auth.!err`	Only priorities info, notice, and warning
`auth.debug;auth.!=warning`	All priorities except warning

Figure 4:

Ações

- O `rsyslog` aceita uma variedade de ações
  - `syslog` tradicional não suporta TCP nem templates
- Um sinal de menos na frente de um nome de arquivo (`-/var/log/messages`) previne o uso de `sync()`

Common actions

Action	Meaning
filename	Appends the message to a file on the local machine
@hostname	Forwards the message to the **rsyslogd** on *hostname*
@ipaddress	Forwards the message to *ipaddress* on UDP port 514
@@ipaddress	Forwards the message to *ipaddress* on TCP port 514
| *fifoname*	Writes the message to the named pipe *fifoname*[a]
user1,user2,…	Writes the message to the screens of *users* if they are logged in
*	Writes the message to all users who are currently logged in
~	Discards the message
^*program;template*	Formats the message according to the *template* specification and sends it to *program* as the first argument[b]

a. See **man mkfifo** for more information.
b. See **man 5 rsyslog.conf** for further details on templates.

Gerando logs na linha de comando

- O comando `logger` permite gerar logs manualmente
  - `-p rec.pri`: especifica recurso e prioridade (default: `user.notice`)
  - `-i`: registra PID
  - `-t xpto`: identifica mensagem com `xpto` (default: username)
- Exemplos

```
### Gera uma mensagem nos logs
$ logger "Hello, logs! (user.notice)"

### Gera msg com nível local3.crit, com tag abcd[PID]
$ logger -p local3.crit -i -t abcd "disco cheio"
```


Rotação de logs: `logrotate`

- `rsyslog` não gerencia rotação de logs
- Existem ferramentas especializadas em rotação
  - `logrotate` é a mais comum no Linux
- O princípio é que periodicamente os arquivos de log são rotacionados, sendo mantidos os últimos N arquivos
  - o arquivo mais antigo (`arq.N`) é apagado, e os demais pulam um número
    - ★ `arq.N-1` → `arq.N`
    - ★ `arq.N-2` → `arq.N-1`
    - ★ `...`
    - ★ `arq` → `arq.1`
  - os arquivos antigos podem ser comprimidos
- Configurado via `/etc/logrotate.conf`

logrotate.conf

- As opções podem ser globais ou por arquivo(s)

logrotate options

Option	Meaning
`compress`	Compresses all noncurrent versions of the log file
`daily`, `weekly`, `monthly`	Rotates log files on the specified schedule
`delaycompress`	Compresses all versions but current and next-most-recent
`endscript`	Marks the end of a `prerotate` or `postrotate` script
`errors` *emailaddr*	Emails error notifications to the specified *emailaddr*
`missingok`	Doesn't complain if the log file does not exist
`notifempty`	Doesn't rotate the log file if it is empty
`olddir` *dir*	Specifies that older versions of the log file be placed in *dir*
`postrotate`	Introduces a script to run after the log has been rotated
`prerotate`	Introduces a script to run before any changes are made
`rotate` *n*	Includes *n* versions of the log in the rotation scheme
`sharedscripts`	Runs scripts only once for the entire log group
`size` *logsize*	Rotates if log file size > *logsize* (e.g., `100K`, `4M`)

Figure 6:

Logs do kernel

- O kernel também gera logs
  - inclusive durante o boot, quando o serviço de logs ainda não iniciou
- Os logs do kernel são armazenados no buffer de anel do kernel (*kernel ring buffer*), um buffer circular de tamanho fixo em memória
  - o buffer pode ser consultado com `dmesg`
  - buffer é exportado via `/proc/kmsg` (R/O) e `/dev/kmsg` (R/W)
- Serviços de log, como `klogd`, `rsyslog` e `systemd-journald`, podem armazenar as mensagens do buffer em arquivos de log

systemd-journald

- Serviço de logs do `systemd`
- Diferente do `syslog` e seus derivados, armazena os logs em formato binário
  - requer ferramentas específicas para acessar os logs
    - `journalctl`, `systemd-cat`
- Na configuração padrão do Ubuntu, as mensagens de log são repassadas pelo `systemd-journal` ao `rsyslog`

journalctl: exemplos (1)

```
### Mostra os logs no journal a partir do início/fim
$ journalctl
$ journalctl -e

### Mostra apenas logs de kernel (equivalente a dmesg)
$ journalctl -k

### Lista boots existentes no journal
$ journalctl --list-boots

### Mostra logs do penúltimo boot (0=atual, -1=último, ...)
$ journalctl -b -2
```

journalctl: exemplos (2)

```
### Mostra os logs no journal a partir de ontem/até 20/09
$ journalctl -S yesterday
$ journalctl -U 09-20

### Mostra os logs entre 10:00 e 12:00
$ journalctl -S "10:00:00" -U "12:00:00"

### Mostra os logs da unidade ssh.service
$ journalctl -u ssh.service
```


journalctl: exemplos (3)

```
### Mostra espaço em disco consumido pelo journal
$ journalctl --disk-usage

### Remove arquivos de journal antigos além de 100 MB
$ journalctl --vacuum-size=100M

### Remove arquivos de journal mais antigos que 2 meses
$ journalctl --vacuum-time=2months

### Limita a 6 arquivos de journal antigos
$ journalctl --vacuum-files=6
```

- `vacuum-*` não remove arquivos ativos (em uso)

Configuração do `systemd-journald`

- O arquivo de configuração é `/etc/systemd/journald.conf`
  - valores default aparecem comentados no arquivo padrão
- A opção `Storage` controla a persistência dos logs
  - `volatile`: apenas memória
  - `persistent`: salva journal em `/var/log/journal`, criando o diretório se não existir
  - `auto`: salva journal em `/var/log/journal` se este diretório existir, caso contrário mantém em memória
    - ★ default
  - `none`: descarta todos os logs
- Opções importantes (`System`=disco, `Runtime`=memória):
  - `SystemMaxUse`, `RuntimeMaxUse`: espaço máximo ocupado
  - `SystemKeepFree`, `RuntimeKeepFree`: espaço que deve ficar livre
  - `SystemMaxFileSize`, `RuntimeMaxFileSize`: tamanho máximo por arquivo

systemd-cat

- Variante de `logger` específica para o `systemd-journald`
  - `logger` funciona tanto com `syslog` quanto com `systemd-journald`
- Exemplos

```
### Gera uma mensagem nos logs
$ echo "Hello, logs! (prioridade=notice)" | systemd-cat

### Gera msg com prioridade crit, com tag abcd
$ systemd-cat -p crit -t abcd echo "disco cheio"
```

107.2 Temporizadores do systemd

Temporizadores (*timers*) do systemd

- Unidades que permitem especificar execução periódica de tarefas
  - equivalente a cron
- O temporizador foo.timer agenda a ativação de uma unidade
  - default é a unidade de serviço foo.service
    - ★ pode ser alterada com Unit
- Um temporizador pode especificar data e horário em termos absolutos (“terças às 13:30”) ou relativos (“15 minutos depois do boot”)

```
### Lista temporizadores ativos
$ systemctl list-timers
```


Tipos de temporizadores

- A unidade de tempo default é segundos, mas é possível especificar outras unidades (anos, meses, dias, horas, minutos, ms, . . .)
- O exemplo abaixo especifica 1a execução em 15 min após o boot e execuções subsequentes com intervalo de 1 dia

```
OnBootSec=15m
OnActiveSec=1d
```

- A opção AccuracySec introduz um atraso aleatório entre 0 e o valor especificado (default 1 minuto)

Type	Time basis
OnActiveSec	Relative to the time at which the timer itself is activated
OnBootSec	Relative to system boot time
OnStartupSec	Relative to the time at which **systemd** was started
OnUnitActiveSec	Relative to the time the specified unit was last active
OnUnitInactiveSec	Relative to the time the specified unit was last inactive
OnCalendar	A specific day and time

Expressões de tempo (OnCalendar)

Time specification	Meaning
2017-07-04	July 4th, 2017 at 00:00:00 (midnight)
Fri-Mon *-7-4	July 4th each year, but only if it falls on Fri–Mon
Mon-Wed *-*-* 12:00:00	Mondays, Tuesdays, and Wednesdays at noon
Mon 17:00:00	Mondays at 5:00 p.m.
weekly	Mondays at 00:00:00 (midnight)
monthly	The 1st day of the month at 00:00:00 (midnight)
*:0/10	Every 10 minutes, starting at the 0th minute
--* 11/12:10:0	At 11:10 and 23:10 every day

Figure 8:

```
### Analisa expressao de tempo
$ systemd-analyze calendar 'mon *-*-* 08,17:05:00'
```

Temporizadores transientes

- Temporizadores associados a comandos, não a unidades `systemd`
  - unidades transientes são criadas automaticamente
- Cobrem a funcionalidade de `at`, mas também podem ser usados com execução periódica não persistente
  - se o sistema for reinicializado, o agendamento é perdido

```
### Executa o comando a cada 10 minutos
$ systemd-run --on-calendar '*:0/10' \
  /bin/sh -c "cd /app && git pull"

### Desliga a máquina às 23:43
$ sudo systemd-run --on-calendar '23:43' poweroff

### Executa cmd daqui a 3 min
$ systemd-run --on-active=3m cmd
```