DEP. ENGª INFORMÁTICA
www.dei.estg.ipleiria.pt

Ficha 3 – `rsyslog, crontab, logrotate`

Tópicos abordados:
- `rsyslog`
- `crontab`
- `logrotate`

1 rsyslog

O `rsyslog` é um serviço pré-instalado e tem por função registar as mensagens e erros do sistema (logs do sistema). O `rsyslog` veio substituir o serviço `syslog`, mas mantém a compatibilidade. O serviço é composto pelos seguintes componentes:

- `rsyslogd` – daemon que assegura as funcionalidades do serviço
- Ficheiros de configuração - `/etc/rsyslog.conf` e `/etc/rsyslog.d/50-default.conf`. O ficheiro de configuração é lido no arranque ou quando o `rsyslogd` recebe o sinal HUP (reiniciar)

```
Ex.: killall -s HUP rsyslogd
```

A estrutura básica do ficheiro de configuração `/etc/rsyslog.conf` é a seguinte:

- **Diretivas globais** – As diretivas globais permitem definir propriedades globais do serviço, como por exemplo o tamanho da fila principal de mensagens. Todas as diretivas globais têm de iniciar uma linha do ficheiro e são precedidas pelo carácter $.
- **Templates** – Permitem definir o formato das mensagens de log.
- **Canais de saída** – Permitem especificar o tipo de saída que um utilizador pretende. Devem ser sempre antes da sua utilização em regras.
- **Regras** – As regras são compostas por seletores e correspondentes ações. Consistem sempre num par composto por seletor e ação, separados por espaços ou tabulações.

Os seletores servem para triar as mensagens através de 2 diretivas, a origem e a prioridade:

- **ORIGEM:** `auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security (ou auth), syslog, user, uucp e local0 a local7.` **Nota**: A origem `security` não deve ser usada por estar obsoleta e a `mark` é apenas usada internamente e não deve ser usada em aplicações.
- **PRIORIDADE** (de menor a maior): `debug`, `info`, `notice`, `config` (ou `warning`, ou `warn`), `err` (ou `error`), `crit`, `alert`, `emerg` (ou `panic`), `none`. **Nota**: `warn`, `error`, e `panic` estão obsoletos; Uma prioridade designa todas as mensagens de prioridade igual ou superior a ela, por exemplo:

```
*.err -> err, crit, alert e emerg
```

A sintaxe pode ainda ser estendida permitindo especificar várias origens na mesma linha:

```
*.=err -> apenas a prioridade err
*.!=err -> todas as prioridades excepto err
mail.emerg;kern.err -> emergências de mail e erros de
                       kernel
```

Quanto ao destino das mensagens, visto como a ação da regra, pode ser (ver Figura 1):

- Ficheiro regular (ex. `/var/log/mesages`)
- Ficheiro especial (ex. `-/var/log/mesages`). Desativa sincronização automática na escrita de mensagens para o ficheiro de log.
- Terminal (ex. `/dev/tty7`)
- Impressora (ex. `/dev/lpt0`)
- Outra máquina -> permite centralizar os logs de sistema (ex. `@192.168.1.1`)

```
# destino ficheiro de texto:
# o '-' omite sincronização imediata
*.debug             /var/log/debug
*.=info;*.notice    -/var/log/messages
# destino terminal e consola:
mail.*              /dev/console
*.warn              /dev/tty7
# destino máquina de rede:
kern.crit           @sounix.estg.ipleiria.pt
# destino lista de utilizadores:
authpriv.crit       root,mantunes
# destino todos os utilizadores logados:
*.crit              *
```

Figura 1 - Exemplo de Ficherio rsyslog.conf

Encontram-se disponíveis para Perl as seguintes funções que permitem a manipulação de mensagens de log: `openlog, syslog,` e `closelog.`

Exemplo de envio de uma mensagem para o log em Perl (consulte `perldoc Sys::Syslog`):

```
# Neste exemplo a origem é 'user' e a prioridade é 'info'
use Sys::Syslog;
openlog($0, 'pid', 'user');
syslog('info', 'este e um teste %d', time);
closelog;
```

Em bash poderemos utilizar o comando `logger` (`man logger`) para escrever para o `rsyslog`.

```
logger –i –s –p local3.warning –t `whoami` "msg via logger"
```

Exercício 1

Recorrendo ao módulo `Sys::Syslog` do PERL, elabore o script `tosyslog.pl` cujo propósito deve ser o envio para o `syslog` da mensagem que é indicada através do primeiro (e único) parâmetro da linha de comando.

2 crontab

O `crond` é um serviço que permite efetuar agendamento de tarefas, efetuado o seu lançamento a horas previamente determinadas.

- Permite automatizar tarefas
- Imitado pelo «`Task Scheduler`» do Windows
    - As saídas do programa (`STDOUT` e `STDERR`) são enviadas por e-mail ao utilizador.
    - `crontab`: contém a configuração utilizada por `crond`
    - cada utilizador dispõe de um crontab pessoal se estiver presente em `cron.allow` e não estiver presente em `cron.deny`
- Comando `crontab`
    - –e - edita a própria crontab
    - –u - «utilizador» → permite a root editar a crontab pessoal de qualquer utilizador
    - -l - lista crontab
    - -r - remove crontab

O editor utilizado por omissão é o "`vi`", que pode ser personalizado usando o seguinte comando: `env EDITOR=nano crontab –e.`

- entradas de comando, com 6 parâmetros:
  «*minutos*» «*hora*» «*dia*» «*mês*» «*dias da semana*» «*comando*»
    - Gamas de valores minutos(0-59)
    - Gamas de valores horas(0-23)

- Gamas de valores dias(1-31)
- Gamas de valores mês(1-12)
- Gamas de valores dias da semana(0-6, 0 →domingo)
- Listas de valores (ex.: 0,8,20 nas horas → meia noite, oito e vinte horas)

- Configuração dos parâmetros:
  - Gamas de valores com saltos (ex.: 0-6 nas horas → todas as horas entre as zero e as seis)
  - Gamas de valores com saltos (ex.: 0-6/2 nas horas → meia noite, duas, quatro, seis)
  - Lista de valores (1,5,10 nas horas → às uma, cinco e dez horas)
  - Um asterisco equivale a gama completa
  - Pode usar abreviaturas (sun, mon, tue, wed, thu, fri, sat)
  - O caracter % representa uma mudança de linha

```
#Exemplos:
00 10 * * mon-fri echo "10 horas, dia da semana"
00 00 * * sat-sun echo "meia noite, fim de semana"
00 20 * * fri echo "20H de sexta-feira"
00 04 * * * echo "4 da manhã"
00 04 * * * tar cvfj /root/etc.`date`.tar.bz2 /etc
```

Exercício 2

a) Com recurso à linguagem Perl, elabore o script "`date.pl`" que deve simplesmente escrever para a saída padrão a data corrente num formato semelhante a 20170220_15h03m23s.

b) Configure o `crontab` do seu utilizador para que o script "`date.pl`" seja executado a cada cinco minutos. A saída produzida pelo script deverá ser redirecionada para o ficheiro "`/tmp/data.txt`".

c) Repita a alínea anterior, mas de modo a que a execução via `crontab` do script "`date.pl`" se realize a cada 2 minutos às 2ª, 4ª e 6ª feiras.

3 logrotate

O `logrotate` é um serviço muito útil, que acompanha normalmente o `rsyslog`. Tem como principal objetivo a rotação dos ficheiros de log, por forma a evitar que estes cresçam desmesuradamente ao ponto de alocarem a totalidade do espaço em disco e assim provocarem a paragem do sistema (DoS).

- Configurações

  - Ficheiro `/etc/logrotate.conf`

  - Directório `/etc/logrotadate.d`

```
#opções de configuração
    rotate 5  # 5 ficheiros de histórico
    weekly #rotação semanal
    compress # compressão dos ficheiros de histórico
```

Mais informação em: `man logrotate`

- Análise dos logs
  - Os logs de pouco servem se não forem analisados!
  - `tail -f <filelog>` - permite visualizar o ficheiro de logs em tempo real. Bastante útil no troubleshooting da configuração de serviços.

O utilitário `logwatch` analisa os logs, produzindo um resumo no modo de execução mais simples, sendo uma ferramenta muito útil para a deteção de anomalias. Não está instalado por omissão no Ubuntu.

```
sudo apt-get install logwatch
```

Deverá igualmente instalar alguns módulos Perl necessários ao logwatch:

```
sudo perl -MCPAN -e 'install Sys::CPU'
sudo perl -MCPAN -e 'install Sys::MemInfo'

#Exemplo de utilização
sudo logwatch --detail 10 --service All | less
```

Exercício 3

a) Altere a configuração do `logrotate` no sentido de guardar por omissão 5 ficheiros de log, de forma comprimida.

b) Altere a configuração do `logrotate` para efetuar a rotação diária dos ficheiros de log no diretório `/tmp/var/abc`, a guardar dois ficheiros de logs com o tamanho mínimo de 1MB.

c) Instale e execute o comando `logwatch`. Interprete os resultados obtidos.

Exercício 4

Implemente uma solução de centralização de logs, através do serviço `rsyslog`. Para tal, ative no servidor central de logs a possibilidade de escrita remota, definindo o parâmetro "`$UDPServerRun`" no ficheiro `/etc/rsyslog.conf`. De seguida teste a escrita remota de mensagens de log através do comando "`logger -n <a.b.c.d>`", sendo `<a.b.c.d>` o endereço IP do servidor central de logs.