NAME

pschecker — keep a watch at running processes

SYNOPSIS

pschecker [**-1**] [**-c**] [**-d**] [**-E** *filename*] [**-e**] [**-i**] [**-o** *filename*] [**-p** *hostname*] [**-q**] [**-R** *time*] [**-s**] [**-t** *timer*] [**-v**] *conffilename*

DESCRIPTION

Given a configuration file and the output of the ps(1) utility, **pschecker** will detect missing processes, running processes that should not at this time, unknown running processes (not declared in configuration file *conffilename*), high elapsed time running processes, virtual size consumers and cpu intensive running processes.

If a condition is realized, depending of the options used, **pschecker** will notify the end user using syslog(3) with LOG_ERR priority (with **-s**), sending UDP datagram to a given host (with **-p** but no **-d**), writing to standard output (with **-d**) or to a file (with **-d** and **-o**).

The **pschecker** utility need to get these fields from the ps(1) utility:

- PID
- %CPU
- ELAPSED
- VSZ
- COMMAND and ARGS

This output is obtained using "/bin/ps -axo pid,pcpu,etime,vsz,command" on FreeBSD, "/usr/bin/ps -eo pid,pcpu,etime,vsz,args" on Solaris and "/bin/ps -eo pid,pcpu,etime,vsz,command" on Linux. Otherwise, you can adapt the code to your wish.

The following options are available:

-1 Run once then quit.

-c Display configuration file then quit.

-d Add more debugging information (multiple **-d** are allowed to increase verbosity). Notifications over network are disabled.

-E *filename*
Read processes to exclude from the file *filename*. They will simply be ignored by the program. The regex(3) interface will be used for the search. Put system processes here. This option is only of interest when **-e** is set (using command line option or directive in configuration file).

-e Detect running processes that are not in configuration file *conffilename*, nor in the list of excluded processes (if **-E** or *#excluded_rule* is used).

-i Output compatible with itrs(1) third party software. Use this option for testing purpose.

-o *filename*
Send output to *filename* instead of stdout.

-p *hostname*
Send UDP datagram to *hostname*.

-q Quiet output if level=0.

-R *time*
Run until *time* (in HHMM format) and display informations about processes. This will help getting a configuration file.

-s Notify the end user using `syslog`(3) with `LOG_ERR` priority.

-t *timer*
Rescan every *timer* second (default 30), if **-1** is not set.

-v Print version and exit.

CONFIGURATION FILE

The configuration is a text file with space separated fields. Each position references a field:

- starting time in HHMM format
- ending time in HHMM format
- options in a comma separated list of *tag=value* pairs, space not allowed
- program name
- program arguments if any, use ... to match until the end, space allowed

A common configuration file:

```
#
#
0000 2400 level=90           vi pschecker.c
0000 2400 level=90,pcpu=80.0 ./a.out ...
0000 2400 level=90,etime=60  xfaces -geometry +10+80
```

The options are:

- *category=string*, this option can be used to assign categories to configuration lines

  ```
  0000 2400 level=90,category=apache /usr/local/sbin/httpd
  ```

- *comment="this is a comment"*, this option can be used to differenciate configuration lines in addition to the **-i** command line option. The first character, the comment delimiter, is required to mark the end but can be escaped at any other position

  ```
  0000 2400 level=90,comment="HTTP server" /usr/local/sbin/httpd
  ```

- *delay=number*, this option will give processes a delay of *number* minutes relative to starting time before being found as missing

- *etime=number*, this option will detect processes with elapsed running time above *number* which is a time representation computed by (hour * 10000 + minutes * 100 + seconds). A special value of 1 will detect processes that started before today. Default is 1 for processes that are not 0000-2400 because they run once a day and 0 otherwise. The value of 0 disable the option

- *level=number*, this option was made to indicate severity from 0 to 100, CRITICAL if above 50, WARNING if above 20 and STANDBY otherwise but this option is just used with **-i** command line option. If number is 0, no error will be output concerning this line, this is useful to allow zero or more processes. If number is 0 and **-q** is used, output concerning this line will be suppressed. Default is 90

  ```
  # 1, 2 or 3 instances of the httpd process can run
  0000 2400 level=90 /usr/local/sbin/httpd
  0000 2400 level=0  /usr/local/sbin/httpd
  0000 2400 level=0  /usr/local/sbin/httpd
  ```

- *mailto=mail@address.com*, if severity level is above 90, add this e-mail address to UDP datagram sent to the probehost (with **-p**)

- *vsz=number*, this option will detect processes whose virtual size is more than *number* KB of memory

- *weekday=7digitsbinarynumber*, this option hides the line for given days in a week. For example, 0111101 refers to Sunday and Friday
- *pcpu=float*, this option will detect intensive cpu processes. The parameter *float* is the percentage cpu usage. Default is 70.0
- *probehost=hostname*, this option can be used to add a probehost for this configuration line only, in addition to the **-p** command line option
- *repeat=number*, this option will replace the need to write the exact same line *number* times in the configuration file

  ```
  0000 2400 level=90 /usr/local/sbin/httpd
  0000 2400 level=0  /usr/local/sbin/httpd
  0000 2400 level=0  /usr/local/sbin/httpd
  ```

  can be replaced by

  ```
  0000 2400 level=90         /usr/local/sbin/httpd
  0000 2400 level=0,repeat=2 /usr/local/sbin/httpd
  ```

Unless escaped, the # character is considered as starting a comment. All characters until the end of the line are removed. For example, if the program name is "prog_#_name", you must use:

```
0000 2400 level=90 prog_\#_name
```

There are also directives you can add to your configuration file, starting at column 1. These directives can override equivalent command line options:

- *#category string*, same as previous description but applies to all lines in the config file
- *#excluded_flag 0* or *#excluded_flag off*, use it to disable the **-e** command line option
- *#excluded_flag 1* or *#excluded_flag on*, same as using **-e** command line option
- *#excluded_rule entry*, add this *entry* to the list of excluded processes. This directive is only of interest when **-e** is set (using command line option or directive in configuration file)

  ```
  # do not check xterm processes on this host
  #excluded_flag on
  #excluded_rule ^xterm$
  ```

- *#excludedprocesscount number*, number of occurrences an excluded process can be present. Default is 100
- *#probehost hostname*, send UDP datagram to this host, this directive is equivalent to the **-p** *hostname* command line option
- *#processes number*, this option will detect an important number of processes. The parameter *number* is the number of processes. Default is 100
- *#sort_by_start_time 1* or *#sort_by_start_time on*, sort process list by start time
- *#sort_by_start_time 0* or *#sort_by_start_time off*, do not sort process list by start time
- *#vsz number*, this option will detect an important use of memory. The parameter *number* is the virtual size in KB. Default is half the memory
- *#weekday 7digitsbinarynumber*, same as previous description but applies to all lines in the config file
- *#quiet_flag 1* or *#quiet_flag on*, same as using **-q** command line option

- *#quiet_flag 0* or *#quiet_flag off*, use it to disable the **-q** command line option

BUGS

Ordering lines in configuration file is important because the first match is chosen according to program name and arguments. It would be very difficult to search for a better match because options (level, pcpu, etime, ...) are not equivalent and can not be ordered.

SEE ALSO

ps(1), regex(3), syslog(3)

AUTHORS

The **pschecker** utility and this manual page were written by Philippe Charnier ⟨charnier@xp11.frmug.org⟩.