

Table of Contents:

Table of Contents:.....	0
Introduction	1
To install SarCheck from CD	2
To install SarCheck from an electronic copy.....	2
To try SarCheck on a test file	3
How to collect data for SarCheck to analyze.....	3
How to activate the software	5
How to order SarCheck	6
How to get the most from SarCheck.....	6
How to run SarCheck (menu driven).....	6
How to change the menu defaults.....	9
How to change SarCheck's algorithms	9
How to run SarCheck (command driven).....	9
How to manually produce sar reports from the command line.....	10
How to set up and create graphs using gnuplot	11
Examples of how to use the switches	11
How to run SarCheck (crontab entry).....	13
How to analyze multiple days of reports	14
How to collect data on one system and analyze it on another	15
Examples for AIX... Collecting data on one system and analyzing it on another.....	15
Examples for Solaris... Collecting data on one system and analyzing it on another.....	16
Examples for HP-UX... Collecting data on one system and analyzing it on another	16
Examples for Linux... Collecting data on one system and analyzing it on another	16
How to move SarCheck to another directory on Unix systems	18
How to access the online instructions / help text.....	18
How to interpret the analysis	19
How to get technical support for SarCheck.....	20
How to get technical support for gnuplot	21
Files included in the SarCheck Version 7 AIX	21
Files included in the SarCheck Version 7 Solaris	22
Files included in the SarCheck Version 7 HP-UX	23
Files included in the SarCheck Version 7 Linux	24
Frequently asked questions (FAQ)	25
How to deinstall SarCheck	29

Note: Additional documentation can be found in the pdf files included in the /opt/sarcheck/doc directory. They give further explanation to an Example SarCheck report, Options available when running 'analyze', and SarCheck parms file keywords.

Introduction

The SarCheck utility analyzes your system for possible performance bottlenecks such as memory shortages and 'leaks', disk load imbalances, CPU bottlenecks, runaway processes, and improperly set tunable parameters. It also tells you approximately how much of an additional workload the system can support at peak times of the day. It does this for UNIX systems by analyzing a user-specified sar report and for Linux systems analyzing data from the /proc filesystem by the sclinagent. The analysis will include ps -elf statistics and the ps data files will include the output from the sarcheckagent to produce a plain English report. The report explains the resource bottlenecks seen and makes recommendations which can improve system performance.

SarCheck will produce graphs if you have a version of gnuplot installed that supports PNG or JPEG output. If you ask for HTML output and the production of graphs, SarCheck will insert the graphs into the HTML document. The graphs will be inserted by using tags complete with a description in the tag's alt attribute in order to meet accessibility requirements.

SarCheck's recommendations are designed to produce incremental improvements, so SarCheck should be run regularly. No attempt is made to guess the ultimately correct value for any parameter based on a single day's analysis. Instead, SarCheck will recommend that you increase or decrease values based on the data available, and will continue to recommend changes until there is no more room for improvement. Performance tuning is, by definition, a process of trial and error. SarCheck will not only help you to make those changes, but will also explain the reasons for each recommendation.

SarCheck is different from other performance tools because it does not monitor system activity. In much the same way that a UNIX performance expert would approach the problem, SarCheck analyzes the data available with sar, ps, /proc filesystem and other utilities you've already have on your system. Since these data collectors are included with the operating system, we didn't see a need to create yet another monitor for you to buy.

SarCheck can be run from the command line, as a cron job, or from a menu-driven front end script. For reasons of safety and security, SarCheck will not attempt to change tunable parameters or anything else in the kernel.

The following conditions are identified by SarCheck Version 7

1. CPU bottleneck detection.
2. I/O bottleneck detection.
3. Detection of improper disk load balancing.
4. Detection of unusually slow disk devices.
5. Memory bottleneck detection.
6. Inefficient sizing of many other tunable parameters.
7. Limited capacity for an increase in workload or users.
8. "Impossible" data, such as negative CPU utilization.
9. Runaway process detection.
10. Memory leak detection.
11. Inefficient PATH variables.
12. Paging space utilization.
13. Analysis of network statistics.

Based on its analysis of the resources and statistics described above, SarCheck may recommend a variety of steps which can be taken to improve system performance.

Known limitations

1. Problems with some AIX implementations of sar are not uncommon. These can frequently be fixed with patches available from IBM.

2. An analysis of the load from an unusually quiet day, such as a holiday, will produce recommendations and capacity planning information that may be inappropriate for days when the system is busy. For this reason, we recommend analyzing activity only from times and days when the system is busy. Please refer to the section entitled "How to produce the most accurate analysis" for details.
3. SarCheck will occasionally report slightly different averages than the sar report, primarily due to rounding errors. SarCheck calculates its own averages because errors are common in the averages calculated by many implementations of sar. For example, the average of the cswch/s column is frequently wrong in the AIX implementation of sar.
4. SarCheck's recommendations are not listed in order of significance or potential for performance improvement.

To install SarCheck from CD

To install the software, log in as root, mount the CD, detar the software and extract the tar archive. You'll need to know the mount point of the CD.

AIX

```
zcat < / (CD mount point) /scaix.taz | tar xvf -  
(works on 32 & 64-bit POWER systems)
```

SOLARIS

```
zcat < / (CD mount point) /scsol64.taz | tar xvf - (64-bit SPARC)  
zcat < / (CD mount point) /scsol32.taz | tar xvf - (32-bit SPARC)  
zcat < / (CD mount point) /scsolx86.taz | tar xvf - (32-bit x86)  
zcat < / (CD mount point) /scsolx64.taz | tar xvf - (64-bit x86)
```

HP-UX

```
zcat < / (CD mount point) /schp.taz | tar xvf -  
(works wherever PA RISC binaries work)
```

LINUX

```
zcat < / (CD mount point) /sclin.taz | tar xvfP -  
(works wherever 32 bit x86 binaries work)
```

To install SarCheck from an electronic copy

In some cases, usually for those with a software subscription, we will email SarCheck to customers. Emailed software is a compressed tar archive. See the section entitled "Files included in this release" for details.

To install the software, log in as root and then uncompress and detar it. **This only takes a few seconds.**

1. Log in as root.
2. Now uncompress & detar:

AIX

```
zcat < scaix.taz | tar xvf -
```

SOLARIS

```
zcat < scsol64.taz | tar xvf - (64-bit SPARC)
zcat < scsol32.taz | tar xvf - (32-bit SPARC)
zcat < scsolx86.taz | tar xvf - (32-bit x86)
zcat < scsolx64.taz | tar xvf - (64-bit x86)
```

HP-UX

```
zcat < schp.taz | tar xvf -
```

LINUX

```
zcat < sclin.taz | tar xvfP -
```

To try SarCheck on a test file

Warning: Do not implement any recommendations produced by analyzing the test files!! They have been included for test purposes only and are located in the /opt/sarcheck/etc directory. To reduce typing, you may want to add /opt/sarcheck/bin to root's PATH.

AIX

```
/opt/sarcheck/bin/analyze -pf /opt/sarcheck/etc/aixps0515 -fs
/opt/sarcheck/etc/aixfs0515 /opt/sarcheck/etc/aixsar15 | more
```

SOLARIS

```
/opt/sarcheck/bin/analyze -pf /opt/sarcheck/etc/solps14 /opt/sarcheck/etc/solsar14 |
more
```

HP-UX

```
/opt/sarcheck/bin/analyze9000 /opt/sarcheck/etc/hpsar22 | more
```

LINUX

```
/opt/sarcheck/bin/analyze /opt/sarcheck/etc/20050221 | more
```

How to collect data for SarCheck to analyze

On UNIX systems, SarCheck gets much of its data from sar. Additional data is collected by the ps -elf command and by our agent. On Linux we do not use sar because the agent collects data from the /proc filesystem. Since this is where the Linux implementation of sar gets its data, the use of sar would only reduce the accuracy of the data that SarCheck analyzes. The crontab settings in these examples will collect data every 10 minutes between 9:00 AM and 5:00 PM, and will delete old files at 6:30 PM.

AIX EXAMPLE

First, **make sure the /var/adm/sa directory exists**, and make it if it doesn't. Uncomment the entries in adm's crontabs file which run the sa1 and sa2 scripts. These can be found in the file /var/spool/cron/crontabs/adm and as a rule, that file should be edited with smit or crontab -e when possible. Once, uncommented, the default entries will capture data once an hour at non-peak times and every 20 minutes during the system's busiest times.

```
#0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3 &
#0 * * * 0,6 /usr/lib/sa/sa1 &
#0 18-7 * * 1-5 /usr/lib/sa/sa1 &
#5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 - ubcwyagvm &
```

The following will change the data collection times, collect more data, and produce a sar report. The crontab settings in this example will collect data every 10 minutes for 24 x 7 monitoring and will delete old files at 11:55 PM. Change adm's crontab to look like this:

```
0 * * * * /usr/lib/sa/sa1 600 6 &
55 23 * * * /usr/lib/sa/sa2 -A &
```

We've created the fsagent that will collect physical and logical volume statistics. The script /opt/sarcheck/bin/fs1 calls the program /opt/sarcheck/bin/bsaixagent. The bsaixagent program collects data from the mount, lsvg and lslv utilities. The fs1 script only needs to be run once and should be run during the times used for SarCheck's analysis. The data file will be located in the /opt/sarcheck/fs directory with the name format of yyymmdd.

Add the following entries to roots' crontab file to collect ps -elf, scaixagent, and fsagent data, and to remove old data files:

```
30 12 * * * /opt/sarcheck/bin/fs1
0,10,20,30,40,50 * * * * /opt/sarcheck/bin/ps1
55 23 * * * /opt/sarcheck/bin/ps2
```

SOLARIS EXAMPLE

First, **make sure the /var/adm/sa directory exists**, and make it if it doesn't. Next, make sure the SUNWaccu and SUNWaccr packages are installed. Finally, uncomment the 3 lines in sys' crontab which run the /usr/lib/sa/sa1 and /usr/lib/sa/sa2 scripts. They should look like this:

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

The following will change the data collection times, collect more data, and produce a sar report. The crontab settings in this example will collect data every 10 minutes for 24 x 7 monitoring and will delete old files at 11:55 PM. Change sys' crontab to look like this:

```
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
55 23 * * * /usr/lib/sa/sa2 -A
```

We've created the scsolstatic agent to collect and save data in a file that can be included in SarCheck's analysis. The data file will be located in the /opt/sarcheck/static directory with the name format of yyymmdd. The script /opt/sarcheck/bin/solst1 calls the program /opt/sarcheck/bin/scsolstatic. The solst1 script only needs to be run once a day and should be run during the times used for SarCheck's analysis. The purpose of the solst2 script is to remove any file in the /opt/sarcheck/static directory which is more than 14 days old. The scsolstatic program saves the disk mapping for disk devices using the c0t0d0 naming convention as well as the sd334 convention used in the sar report. If the -ctd switch or the CTD sarcheck_parms keyword is part of your analysis the scsolstatic agent should be included in the crontab entries.

For os releases prior to Solaris 10 add the following entries to roots' crontab file if they don't exist. Solaris 10 releases add the entries to sys' crontab file. These entries will collect ps -elf, scsolagent, scsolstatic data and remove old data files.

```
0,10,20,30,40,50 * * * * /opt/sarcheck/bin/ps1
55 23 * * * /opt/sarcheck/bin/ps2
5 12 * * * /opt/sarcheck/bin/solst1
45 12 * * * /opt/sarcheck/bin/solst2
```

HP-UX EXAMPLE

First, **make sure the /var/adm/sa directory exists**, and make it if it doesn't. Add the following to your /usr/spool/cron/crontabs/root file using SAM or crontab -e to collect sar, ps, and schpagent data. The crontab settings in this example will collect data every 10 minutes for 24 x 7 monitoring and will delete old files at 11:55 PM.

```
0,10,20,30,40,50 * * * * /usr/lbin/sa/sa1
55 23 * * * /usr/lbin/sa/sa2 -A
0,10,20,30,40,50 * * * * /opt/sarcheck/bin/ps1
55 23 * * * /opt/sarcheck/bin/ps2
30 12 * * * /opt/sarcheck/bin/vg1
```

The vg1 script calls the /opt/sarcheck/bin/vgparse program. The vgparse program collects data from the vgdisplay and pvdisplay programs. This data is used by SarCheck to find disks which are inefficient because of fragmented logical volumes. The vg1 script only needs to be run once and should be run at a time when the data will be included in SarCheck's analysis.

Note: If you have edited the crontab entries with vi instead of crontab -e, it may be necessary to reboot the system for the changes to take effect. The ps -elf data collected on large systems can take up a considerable amount of space. If you want to store this data somewhere other than /opt/sarcheck/ps, you can specify a different directory with the PSELFDIR keyword in the sarcheck_parms file.

WARNING: If you choose to specify a different directory, be sure to pick a directory that is not used for anything else. The purpose of the ps2 script is to remove any file in the ps -elf directory which is more than 14 days old and you don't want to accidentally remove files which contain something other than ps -elf data.

LINUX EXAMPLE

SarCheck for Linux uses the sclinagent to collect data and stores the files in the /opt/sarcheck/procstat directory. The data files use the yyyyymmdd naming convention. Additional data is collected from ps -elf with our ps1 script. The crontab settings in this example will collect data every 10 minutes for 24 x 7 monitoring and will delete old files at 11:55 PM. To start collecting data, su as sys (or root if necessary) and add the following lines to that user's crontab file.

```
0,10,20,30,40,50 * * * * /opt/sarcheck/bin/prst1
55 23 * * * /opt/sarcheck/bin/prst2
0,10,20,30,40,50 * * * * /opt/sarcheck/bin/ps1
55 23 * * * /opt/sarcheck/bin/ps2
```

Resource utilization statistics are collected with the prst1 and ps1 scripts. Old files are deleted with the prst2 and ps2 scripts.

How to activate the software

Evaluation and licensed software will require an activation key. Run analyze -s and forward the output to sales@sarcheck.com. Feel free to install eval software on as many systems as you want and use it until it expires.

SarCheck was not designed to be regularly moved from one system to another, in an effort to provide "quick fixes" to a number of systems. Quick fixes will not allow you to take advantage of the long term iterative tuning that SarCheck makes possible.

How to order SarCheck

Use the -o option of /opt/sarcheck/bin/analyze or the -o option of /opt/sarcheck/bin/analyze9000 for HP-UX to produce an order form. You can email or fax it to us, or ask your reseller to contact us. SarCheck's pricing is the same throughout the world. The cost of shipping SarCheck via US Mail is included in the price of SarCheck. If you'd like the software shipped via Federal Express, DHL, etc., please provide your account number and we will be happy to accommodate you.

In some parts of the world, local resellers may charge prices which are higher than our list price because they pay for the currency conversions, international shipping, duties, support, etc. We urge our customers to support their resellers.

How to get the most from SarCheck

To get the most benefit out of SarCheck, we recommend using it as follows:

- 1. Analyze data from the time of day when performance is most important.** SarCheck looks at an individual day of data reported by sar or the Linux procstat data file. That data should reflect the times of the day when performance is most important, ideally on the most active days of the week or month. Select the times of day that you want to analyze by using a sar report from that time period, or by using the -st and -en switches to analyze a subset of the data in the sar report. These switches can be used with either the sarcheck script or the analyze program. For example, if you want to analyze data from 7:00 to 16:00, use `sarcheck -st 07 -en 16` to specify the starting and ending times.
- 2. Review the recommendations based on several days of statistics;** especially days when peak processing occurs, and implement the recommendations that occur consistently.
- 3. Don't implement all of the recommendations at the same time.** All performance tuning involves trial and error and every system is unique, therefore some recommendations may occasionally hurt performance. This is an uncommon occurrence, but these unsuccessful attempts to improve performance can only be identified if recommendations are implemented individually. Some tunable parameter recommendations should be implemented together (for example, AIX maxperm and minperm) and SarCheck will tell you when you need to do this.
- 4. Continue using SarCheck on a regular basis.** While SarCheck will probably make recommendations when it is first run, that's only the beginning. Since many of the changes recommended by SarCheck are small, they will gradually lead you towards a truly optimized system. As new users and applications are added, older programs are modified, and file sizes increase, SarCheck will help to keep you from being surprised by the changing demands of your applications and users over a period of months or years. The evaluation period is designed to get you started, but it can not provide the long term benefit that SarCheck's algorithms were designed for.

How to run SarCheck (menu driven)

First, log in as root. To analyze sar statistics from a menu, type:

```
/opt/sarcheck/bin/sarcheck
```

To reduce typing, you may want to add /opt/sarcheck/bin to root's PATH.

A series of choices will appear on the screen. If you accept all the defaults by pressing the Enter key, the previous day's sar data will be analyzed, and this is the easiest way to get started. For security reasons, your account must have permission to

access the sar data or report files that you wish to analyze.

The first question will ask you whether you want to analyze sar data or a sar report. Sar data is usually found in `/var/adm/sa/sann`, where *nn* is the day of the month. Sar reports are already reduced into a readable form and are usually found in `/var/adm/sa/sar nn` .

Analyzing the reports will be marginally faster than analyzing the data, but an advantage to analyzing the data is that you can control the start and end times by changing SarCheck's defaults. To change any of the defaults, see the section "How to change the SarCheck menu defaults".

```
Analyze what?      d   sar data (sa files)
                   r   A sar report (sar files)
                   c   Concatenate existing sar reports
                   *   Accept all defaults
                   x   Exit SarCheck
(keyword = DR, default = d): _
```

After you pick the `d` or `r` options, you will be prompted to enter the name of the data or report file. In either case the default will be the statistics from the previous business day. The `c` option will concatenate all of the sar reports present and will not ask you for the name of a file. The `sarcheck` script will change your working directory, so you do not have to use the absolute address of the file. To accept all defaults, enter an asterisk; to exit `sarcheck`, enter an `x`. To change any of the defaults, see the section "How to change the SarCheck menu defaults".

Sar data is usually found in `/var/adm/sa/sann`. Based on user-definable defaults, data from 08:00 to 17:00 will be analyzed. Enter the name of the sar data file that you wish to analyze.

```
Available data files in /var/adm/sa:
sa20, sa21, sa25, sa26, sa27, sa28, sa29, sa30
(default = sa29):
```

Note that if you run `sarcheck` on a Saturday, Sunday or Monday, Friday's statistics will be analyzed by default. This is because weekend statistics are usually not representative of a "loaded" system, and there is a possibility that misleading recommendations would be generated. To change the default of excluding the analysis of weekend data, see the section "How to change the SarCheck menu defaults".

The next option allows you to pick formatting. The default will produce a report with page numbers and page breaks (`ctrl-L`) included. For users that prefer to paginate the report with another tool, such as `pg`, the `p` option will suppress these page breaks. You can also choose to produce an HTML document at this point, and can decide whether to format the disk analysis in table form. HTML documents are best viewed with a web browser. If you wish to exit `sarcheck`, enter an `x`.

```
Pick formatting:  n   Normal, with page breaks
                 p   Page breaks suppressed
                 h   Create HTML document, no disk
                   table
                 t   Create HTML document, with disk
                   table
                 o   Create HTML document, disk
                   table only
                 *   Accept remaining defaults
                 x   exit sarcheck
(keyword = OPT, default = n): _
```

The verbosity option controls how verbose the SarCheck report is. The default verbose mode may produce a report 5 pages long, while superquiet mode may only contain 5 lines of text. Please note that instructions for implementing recommendations, explanations, and alternate tuning strategies may be suppressed by the quiet modes. When you're first using SarCheck, we recommend using the verbose mode so that you don't miss anything. The superquiet mode will automatically suppress page breaks.

```
Verbosity level:  v   Verbose mode
                 q   Quiet, most verbiage
                   suppressed
```

```

        Q   Superquiet, all verbiage
            suppressed
        *   accept all defaults
        x   Exit SarCheck
(keyword = VERBOSE, default = v): _

```

Analysis of ps -elf data will provide you with a closer look at memory bottlenecks and the ability to detect runaway processes and memory leaks. The enhanced sensitivity option increases the probability of generating "false alarms". See the section entitled "Setup of ps -elf data collection" for more information. If you wish to exit sarcheck, enter an x.

```

Analyze ps -elf  n   No, analyze sar data only
data?           y   Yes, analyze sar and ps
                e   Enhanced sensitivity of
                ps data analysis
                *   accept all defaults
                x   exit sarcheck
(keyword = PSELFOPT, default = y): _

```

The disk filtration option is useful for large systems and has no effect on smaller systems. If more than 12 disks are present, sarcheck will only print a paragraph on each disk which has significant activity. SarCheck will try to determine what "significant" means on your system. If you find that SarCheck's filtration is too aggressive, use "z" to show all disks with some activity. We have discovered that on systems with several thousand disk devices, it is common for most of them to be completely idle. The "z" option will filter out only disks with no activity at all.

```

Disk filtration:  y   Filter disk analysis more than 12
                  disks.
                  n   Analyze all disk activity if more
                  than 12 seen.
                  z   Analyze all disks with some
                  activity.
                  *   Accept all defaults
                  x   Exit sarcheck
(keyword = DISKFLTR, default = y):

```

The tabular summary is used to print a summary of statistics in table form at the end of the report, and an example is included later in the manual. If HTML output has been selected, an HTML table is created. This option is useful for transferring statistics to a spreadsheet or graphics program, producing output which can be easily parsed by other programs. and for generating an easy to read table at the end of an HTML page.

```

Tabular Summary? y   Print a tabular summary at the
                  end of the report
                  i   Print a tabular summary instead of
                  the report
                  n   Print the report without a summary
                  *   Accept remaining defaults
                  x   exit sarcheck
(keyword = TABULAR, default = y):

```

This option is used to decide where to send the analysis. Note that some of these choices will be different based on the pager you use and any modifications made to the defaults.

If you choose to send the output to a file, you'll be prompted for the name of the file. The default file name is /tmp/yyymmddhhmmss, which is a date/time stamp. You can modify the default by editing the sarcheck script. If you wish to exit sarcheck, enter an x.

```

Send output to:  1   more (the screen)
                  2   lp -s (a printer)
                  3   A file
                  x   exit sarcheck
(keyword = OUTOPT, default = 1): _

```

How to change the menu defaults

The sarcheck script will look for the file named `/opt/sarcheck/etc/sarcheck_parms` and will use any values found there instead of the normal defaults. This file is not included as part of the SarCheck distribution and you'll need to create it if you want to use it. The syntax for the `sarcheck_parms` file is very simple. Create a line in the file with the keyword and its new default value, separated by a space.

The keywords for SarCheck's menu options can be found when running the `sarcheck` script, and the value should be one of the choices on the menu. For example, if you want the output of SarCheck to include a tabular summary at the end of the report, here is the menu selection that you will see:

```
Tabular Summary?  y  Print a tabular summary at the
                   end of the report
                   i  Print a tabular summary instead of
                       the report
                   n  Print the report without a summary
                   *  Accept remaining defaults
                   x  exit sarcheck
(keyword = TABULAR, default = n):
```

You can see the name of the keyword and the options available. To change the default from 'n' to 'y' for this menu item, add the following line to the `sarcheck_parms` file:

```
TABULAR y
```

Now when you run the `sarcheck` script, the default behavior will be to print a tabular summary at the end of the report. After those two fields are parsed by the `sarcheck` script, the rest of the line is ignored and is available as a comment. Any line that starts with something other than a valid keyword is also treated as a comment and is ignored.

Once you have decided to change the defaults, create or edit the `sarcheck_parms` file. Here is an example of a `sarcheck_parms` file where the starting and ending times used for analysis have been changed, page numbering is suppressed, and a tabular summary is printed at the end of the report. Note that since the `sarcheck` script only looks at the first two fields on each line, the rest of the line is treated as a comment and lines that don't start with valid keywords are also treated as comments:

```
# file to customize sarcheck created by
# Jess the sys admin on March 23, 2008
#
ST 06:00  starting time is 6AM
EN 15:00  ending time is 3PM
OPT p     suppress page numbering
TABULAR y add a tabular summary
GRAPHDIR /diskfarm/sarcheck/images
```

How to change SarCheck's algorithms

The `sarcheck_parms` file can also be used to change the thresholds used by SarCheck's tuning algorithms. Default values of SarCheck's thresholds have been established based on feedback from hundreds or thousands of systems and these values should not be overridden without good reason. A complete list of keywords supported in the `sarcheck_parms` file can be found in SarCheck `parms` file `keywords.pdf` located in `/opt/sarcheck/doc` directory.

How to run SarCheck (command driven)

AIX EXAMPLE

To analyze a sar report file called sar15 file type:
`/opt/sarcheck/bin/analyze sar15`

SOLARIS EXAMPLE

To analyze a sar report file called sar12, type:
`/opt/sarcheck/bin/analyze sar12`

HP-UX EXAMPLE

To analyze a sar report file called sar12, type:
`/opt/sarcheck/bin/analyze9000 sar12`

LINUX EXAMPLE

To analyze a `/opt/sarcheck/procstat` data file called `/opt/sarcheck/procstat/20050221`, type:
`/opt/sarcheck/bin/analyze -ps /opt/sarcheck/procstat/20050221`

To reduce typing, you may want to add `/opt/sarcheck/bin` to root's PATH.

For best results, pipe the output to `more` so that you can read it, or redirect it to a file if you want to save it. A report will be produced which contains information about your system, a brief summary, a recommendations section (if applicable), a resource analysis section, and a capacity planning section (if not suppressed). For more information, see the section entitled "How to Interpret the Analysis".

How to manually produce sar reports from the command line

For UNIX users you may want to create a new sar report from an existing sa data file:

If your system is in use from 8:00AM to 5:00PM, and runs batch jobs and backups at night with plenty of time to spare, performance is probably most important during the day. To help SarCheck produce the most useful analysis, use a command such as:

```
sar -A -s8 -e17 > reportfile
```

This command will use all options (-A), and will only include data collected between 8:00AM (-s8) and 5:00PM (-e17) in the report. If the busiest day in the last few weeks was the 29th of the month, and you want to produce a report of system activity between 8:00 and 5:00 on that day, use the following command:

```
sar -A -s8 -e17 -f /var/adm/sa/sa29 > report29
```

Of course, this will only work if `/usr/adm/sa/sa29` actually exists. Sar data should be regularly collected by a crontab entry.

SarCheck does not need (and can not use) sar reports produced with sar's -P switch. Please do not use the -P switch when producing reports for SarCheck to analyze.

How to set up and create graphs using gnuplot

If you ask SarCheck to produce PNG or JPEG graphs, it will look for gnuplot and will try to use it to generate graphs. There are a number of different places on the World Wide Web where you can find gnuplot and it is generally available as source code or source code with a precompiled binary.

For the sake of consistency and convenience, we use the precompiled binary built by Ready to Run Software, Inc. It is available on our web site at <http://www.sarcheck.com/gnuplot/>

To set up gnuplot, please follow the instructions on the web site.

SarCheck can build graphs with the gnuplot utility. By adding the `-png`, `-jpg`, or `-jpeg` switches, SarCheck can use gnuplot to produce PNG or JPEG graphs and insert those graphs in the HTML output of SarCheck. This will enable you to post some really interesting SarCheck reports on your corporate intranet. To produce an HTML report with PNG graphs, use the `-html` and `-png` switches when running `analyze`. The command will use 20080512 and `sar12` files as an example.

```
For AIX and Solaris systems:   analyze -html -png sar12 > rpt12.html
For HP-UX systems:           analyze9000 -html -png sar12 > rpt12.html
For Linux:                   analyze -html -png /opt/sarcheck/procstat/20080512 > rpt12.html
```

This will produce an HTML report which can be read by your favorite browser. The most important parts of the report will be printed in bold type and headings are used to clarify what you're looking at. Graphs are inserted in appropriate places in the body of the report and some additional text is added to help explain the significance of the graphs. For more information, for AIX see "Options available when running 'analyze'" pdf in the `/opt/sarcheck/doc` directory.

If you want SarCheck to produce graphs without the accompanying SarCheck report, use the `-gonly` switch to produce "graphs only".

Please go to <http://www.sarcheck.com/graphhints.htm> for helpful graphing hints.

Examples of how to use the switches

The number of switches and options available in SarCheck continues to grow, and this section is designed to help you decide how to do what you want. In order to maintain some level of clarity, all examples will analyze the sar report file `/var/adm/sa/sar23` and include `ps -elf` in the analysis. The output of the `analyze` program is stdout, so you'll probably want to pipe it to more or redirect it to a file.

Note: The syntax for these examples shown are for the Solaris and AIX platforms. HP-UX would differ by using the `analyze9000` command instead of `analyze`. Linux users would substitute the `/opt/sarcheck/procstat` data file for the `sar23` file.

Example 1: Analyzing a sar report and ps -elf output. We're going to start with a simple example. The command below will run the `analyze` program and tell it to analyze the sar report file `/var/adm/sa/sar23` in conjunction with data collected by the `/opt/sarcheck/bin/ps1` script. The `-ps` switch tells the `analyze` program to search for the `ps -elf` data and include it in the report if possible. The `ps` data file also includes various kernel statistics collected by our agent to produce a more comprehensive analysis.

```
analyze -ps /var/adm/sa/sar23
```

Example 2: Removing the page breaks. The `-p` switch removes the page breaks. This is especially useful when piping the

report to pg instead of more. A number of other switches, the -html switch for example, will automatically invoke -p where it makes sense.

```
analyze -ps -p /var/adm/sa/sar23
```

Example 3: Specifying a different ps -elf file. If you move the sarcheck files to a directory other than /opt/sarcheck and you want to analyze a particular ps -elf data called pselffile, you have to tell the analyze program where to find the data and the new name. This example looks at the sar report sar23 and the ps -elf data in pself file:

```
analyze -pf /tmp/pselffile /var/adm/sa/sar23
```

Example 4: Specifying a different ps -elf directory. If you move the sarcheck files to a directory other than /opt/sarcheck and you want to analyze ps -elf data, you have to tell the analyze program where to find the data. This example looks at the sar report sar23 and the ps -elf data in the /test directory:

```
analyze -pd /test /var/adm/sa/sar23
```

Example 5: Creating an HTML-formatted report. Let's combine a few switches this time. The -html switch makes the -p switch unnecessary, but we want to incorporate ps -elf data into the analysis and we want to see a quick table of some statistics at the end of report.

```
analyze -html -ps -t /var/adm/sa/sar23
```

Example 6: Creating an HTML-formatted report with embedded graphs. This is the same as the previous example, except that graphs are now embedded in the HTML output and they will be visible when the output is viewed with a browser. For this to work properly, you must have a copy of gnuplot installed, SarCheck must be able to find it, and the output must be redirected to a file. The file is then opened with a browser that can display PNG graphs. You can also use the -jpeg or -jpg switches if your version of gnuplot supports jpeg output.

```
analyze -html -ps -t -png /var/adm/sa/sar23 > sar23.html
```

Example 7: Creating an HTML-formatted report, but this time on a big system. This time we're analyzing sar data from a large system, and there are 100 disks on the system. By default, SarCheck will filter out information on 'uninteresting' disks, but it will still produce a paragraph on each disk. This can get a little ridiculous and hard to read, so we'll use the -dtbl switch to format the disk information into an HTML table, and the -dbusy switch to sort the disk information so that the busiest disks are at the top of the table. The -ptbl switch is also being used to format ps -elf statistics into a table. As with the -t switch, cells will be colored if SarCheck wants to draw your attention to specific data. Note that the -dtbl and -ptbl switches are most useful with the -html switch.

```
analyze -html -ptbl -t -dtbl -dbusy /var/adm/sa/sar23
```

Example 8: Emailing the output. If you manage a network of 200 systems, you may want to email interesting SarCheck reports to yourself, but you probably don't want to be spammed with 200 messages a day saying that everything's okay. The -r switch prevents SarCheck from producing any output at all if there are no recommendations. The -Q switch reduces verbiage to a minimum.

```
analyze -ps -r -Q /var/adm/sa/sar23 | mail root@wherever.com
```

Example 9: Suppressing the detection of memory leaks. False alarms are not uncommon when SarCheck attempts to detect memory leaks. Some programs, such as those found on systems running Oracle, will grow over time. This resembles a deliberate memory leak, and depending on the behavior of programs running on your system, you may want to suppress the reporting of memory leaks, runaway processes, or unusually large processes. The -pml switch can be used to suppress memory leaks as follows:

```
analyze -ps -pml /var/adm/sa/sar23
```

Example 10: Analyzing ps -elf output in conjunction with the sar report. The /opt/sarcheck/bin/ps1 script is used to collect ps -elf data which can be used in conjunction with the sar report. The -ps switch tells the analyze program to search for the ps -elf data and include it in the report if possible.

```
analyze -ps /var/adm/sa/sar23
```

Example 11: Creating a SarCheck HTML report with graphs and moving the files to another system

UNIX Example:

1. Pick or create an empty directory to hold the HTML and graphs. For this example, we'll use the directory /tmp/sarcheck.
2. Create a SarCheck analysis of a sar report called sarxx.
3. Run the following command with these switches to put all of the pieces in /tmp/sarcheck, and put usable graph names in the HTML file's IMG tags:

```
analyze -png -html -gd /tmp/sarcheck -hgd ./ -t -ps sarxx > /tmp/sarcheck/test.html
```

Note: HP-UX users substitute analyze9000 for analyze.

The HTML file is redirected to /tmp/sarcheck, the -gd switch puts the graphs in /tmp/sarcheck, and the -hgd switch puts “.” in the IMG tags of the HTML report instead of the actual directory name. As long as the graphs and the HTML file are in the same directory, the browser will have no trouble finding the graphs.

4. Move the files in /tmp/sarcheck with FTP or whatever tool you prefer. Be sure that the graphs are transferred as “binary” and not “text”. This is a common problem, especially when a PC running Windows is in use.

LINUX Example:

1. Pick or create an empty directory to hold the HTML and graphs. For this example, we'll use the directory /tmp/sarcheck.
2. Create a SarCheck analysis of a /opt/sarcheck/procstat data file called 20070526 (remember, this is just an example and you might not find this file on your system.)
3. Run the following command with these switches to put all of the pieces in /tmp/sarcheck, and put usable graph names in the HTML file's IMG tags:

```
analyze -png -html -gd /tmp/sarcheck -hgd ./ -t -ps 20070526 > /tmp/sarcheck/test.html
```

The HTML file is redirected to /tmp/sarcheck, the -gd switch puts the graphs in /tmp/sarcheck, and the -hgd switch puts “.” in the IMG tags of the HTML report instead of the actual directory name. As long as the graphs and the HTML file are in the same directory, the browser will have no trouble finding the graphs.

4. Move the files in /tmp/sarcheck with FTP or whatever tool you prefer. Be sure that the graphs are transferred as “binary” and not “text”. This is a common problem, especially when a PC running Windows is in use.

How to run SarCheck (crontab entry)

SarCheck can be run automatically by adding an entry to root's crontab file, ideally using crontab -e. Determine the time that /usr/lib/sa/sa2 is run (you'll have to set this up), and use cron to run /opt/sarcheck/bin/analyze after that time. Here are two examples which assume sa2 is run at 18:00, the analysis will be done at 18:05:

In order to print a SarCheck analysis every weeknight, use the following entry:

AIX and SOLARIS EXAMPLE

```
5 18 * * 1-5 /opt/sarcheck/bin/analyze /var/adm/sa/sar`date +%d` | lp -s
```

To keep all of SarCheck's recommendations in the /usr/ops directory, use the following entry:

```
5 18 * * 1-5 /opt/sarcheck/bin/analyze /var/adm/sa/sar`date +%d` > /usr/ops/`date +%y%m%d`
```

HP-UX EXAMPLE

```
5 18 * * 1-5 /opt/sarcheck/bin/analyze9000 /var/adm/sa/sar`date +%d` | lp -s
```

To keep all of SarCheck's recommendations in the /usr/ops directory, use the following entry:

```
5 18 * * 1-5 /opt/sarcheck/bin/analyze9000 /var/adm/sa/sar`date +%d` > /usr/ops/`date +%Y%m%d`
```

LINUX EXAMPLE

SarCheck can be run automatically by adding an entry to the user's crontab file, ideally using crontab -e. Here is an example which assumes the analysis will be done at 18:05.

In order to print a SarCheck analysis every weeknight, use the following entry:

```
5 18 * * * /opt/sarcheck/bin/analyze /opt/sarcheck/procstat/`date +%Y%m%d` | lp -s
```

To keep all of SarCheck's recommendations in the /usr/ops directory, use the following entry:

```
5 18 * * * /opt/sarcheck/bin/analyze /opt/sarcheck/procstat/`date +%Y%m%d` >/usr/ops/`date +%Y%m%d`
```

Because the output of the analyze program is stdout, you can pipe or redirect it in lots of ways. It can be printed, mailed, stored... whatever works best in your environment.

How to analyze multiple days of reports

SarCheck has the ability to analyze multiple days of data at once when the reports are concatenated. The only limitation is that these reports must actually exist and be valid.

UNIX EXAMPLE

Here is an example of how to analyze all sar reports from the first seven days of the month. **Note:** HP-UX users substitute analyze9000 for analyze.

1. First concatenate the reports, creating a single report called /tmp/multisar:

```
cat /var/adm/sa/sar0[1-7] > /tmp/multisar
```
2. Now analyze the concatenated report and pipe it to more:

```
/opt/sarcheck/bin/analyze /tmp/multisar | more
```

Once you've become used to working with concatenated sar reports, you'll probably discover that the find command in /usr/lib/sa/sa2 removes the sar reports too quickly, and the naming convention used in /usr/lib/sa/sa1 is too restrictive. Make copies of the original sa1 and sa2 scripts, and then modify them to meet your needs.

LINUX EXAMPLE

Here is an example of how to analyze data from the first seven days of March.

1. First concatenate the reports, creating a single report called /tmp/multi:

```
cat /opt/sarcheck/procstat/2004030[1-7] > /tmp/multi
```
2. Now analyze the concatenated report and pipe it to more:

```
/opt/sarcheck/bin/analyze /tmp/multi | more
```

Please note: The analyze program does not work if wildcard characters are used as a filename. Wildcard characters should be used with the cat command in order to produce a single file for the analyze program.

How to collect data on one system and analyze it on another

Version 7 of SarCheck is now able to collect and save information from the current system so the analysis can take place on a different system. The scaixagent on AIX , scsolagent on Solaris and schpagent on HP-UX will collect data and the output of various commands will be stored in the data files found in /opt/sarcheck/ps. For AIX, the /opt/sarcheck/bin/vmsparse program has been replaced with /opt/sarcheck/bin/scaixagent. For Solaris, the /opt/sarcheck/bin/dnlcmon program has been replaced with /opt/sarcheck/bin/scsolagent. For HP-UX, the /opt/sarcheck/bin/freemem program has been replaced with the program /opt/sarcheck/bin/schpagent.

If the analysis is to take place on a different system you will need to bring the applicable sar report and ps files to be analyzed. It is important to make sure the system where the analysis is taking place has a unique naming convention for the non-native sar and ps data files. This will prevent inadvertently overwriting one system's data with another, since the default sar report name /var/adm/sa/sardd and ps data file name of /opt/sarcheck/ps/yyyyymmdd could exist on multiple systems.

On AIX systems, if the analysis is to take place on a different system you will also need to bring the data created by the fsagent /opt/sarcheck/fs/yyyyymmdd , so the filesystem statistics be included in the analysis.

Examples for AIX... Collecting data on one system and analyzing it on another

The data files you need to bring are the sar, ps and fs (filesystem) data files.

1. Create directories for the data files.

```
sar      /var/systemname/sar
ps       /var/systemname/ps
fs       /var/systemname/fs
```

2. Create directories for the graphs and html output.

```
graph images  /var/systemname/images
html output   /var/systemname/output
```

Example A (AIX):

This example will use the analyze command line with switches to point to the data files in the new directories. The systemname will be called skippy and the data is for May 5, 2008. It will produce a report with html,png images, text and tables of disk, filesystem, and ps elf statistics. It includes increasing the default width of the graphs, putting the graphs in the images directory and including the -hgd switch for the browser to locate the graphs. The following command should all be on one line.

```
analyze -dtoo -ptoo -wide -png -t -html -pd /var/skippy/ps -gd /var/skippy/images -hgd ../images
/var/skippy/sar/sar05 > /var/skippy/ouptut/20080505.html
```

Example B (AIX):

Another example is to use the /opt/sarcheck/etc/sarcheck_parms file. The sarcheck_parms file is a flat file that will change the default behavior of the analyze program. You can include specific keywords that will shorten the analyze command line.

```
PSELFDIR /var/skippy/ps
FSDIR    /var/skippy/fs
GRAPHDIR /var/skippy/images
WIDE
HTMLGRAPHDIR ../images
```

```
analyze -png -dtoo -ptoo -t -html /var/skippy/sar/sar05 > /var/skippy/output/20080505.html
```

Examples for Solaris... Collecting data on one system and analyzing it on another

The data files you need to bring are the sar, ps and static files.

1. Create directories for the data files.

```
sar      /var/systemname/sar
ps       /var/systemname/ps
static   /var/systemname/static
```

2. Create directories for the graphs and html output.

```
graph images  /var/systemname/images
html output   /var/systemname/output
```

Example A (Solaris):

This example will use the analyze command line with switches to point to the data files in the new directories. The systemname will be called skippy and the data is for September 5, 2008. It will produce a report with html, png images, text and tables of disk, and ps elf statistics. It includes increasing the default width of the graphs, putting the graphs in the images directory and including the -hgd switch for the browser to locate the graphs. The following command should all be on one line.

```
analyze -dtoo -ptoo -wide -png -t -html -pd /var/skippy/ps -gd /var/skippy/images -hgd ../images
/var/skippy/sar/sar05 > /var/skippy/ouptut/20080905.html
```

Example B (Solaris):

Another example is to use the /opt/sarcheck/etc/sarcheck_parms file. The sarcheck_parms file is a flat file that will change the default behavior of the analyze program. You can include specific keywords that will shorten the analyze command line.

```
PSELFDIR /var/skippy/ps
GRAPHDIR /var/skippy/images
WIDE
HTMLGRAPHDIR ../images
STATICDIR /var/skippy/static
```

```
analyze -png -dtoo -ptoo -t -html /var/skippy/sar/sar05 > /var/skippy/output/20080905.html
```

Examples for HP-UX... Collecting data on one system and analyzing it on another

The data files you need to bring are the sar and ps data files.

1. Create directories for the data files.

```
sar      /var/systemname/sar
ps       /var/systemname/ps
```

2. Create directories for the graphs and html output.

```
graph images  /var/systemname/images
html output   /var/systemname/output
```

Example A (HP-UX):

This example will use the analyze command line with switches to point to the data files in the new directories. The systemname will be called skippy and the data is for September 5, 2008. It will produce a report with html, png images, text and tables of disk, and ps elf statistics. It includes increasing the default width of the graphs, putting the graphs in the images

directory and including the -hgd switch for the browser to locate the graphs. The following command should all be on one line.

```
analyze9000 -dtoo -ptoo -wide -png -t -html -pd /var/skippy/ps -gd /var/skippy/images -hgd ../images  
/var/skippy/sar/sar05 > /var/skippy/ouptut/20080905.html
```

Example B (HP-UX):

Another example is to use the /opt/sarcheck/etc/sarcheck_parms file. The sarcheck_parms file is a flat file that will change the default behavior of the analyze program. You can include specific keywords that will shorten the analyze command line.

```
PSELFDIR /var/skippy/ps  
GRAPHDIR /var/skippy/images  
WIDE  
HTMLGRAPHDIR ../images  
STATICDIR /var/skippy/static (For Solaris)
```

```
analyze9000 -png -dtoo -ptoo -t -html /var/skippy/sar/sar05 > /var/skippy/output/20080905.html
```

Examples for Linux... Collecting data on one system and analyzing it on another

The data files you need to bring are the /opt/sarcheck/procstat and /opt/sarcheck/ps data files. Please note both files use the same naming convention of yyymmdd.

1. Create directories for the data files.

```
procstat /var/systemname/procstat  
ps /var/systemname/ps
```

2. Create directories for the graphs and html output.

```
graph images /var/systemname/images  
html output /var/systemname/output
```

Example A (Linux):

This example will use the analyze command line with switches to point to the data files in the new directories. The systemname will be called skippy and the data is for May 5, 2011. It will produce a report with html png images, text and tables of disk and ps elf statistics. It includes increasing the default width of the graphs, putting the graphs in the images directory and including the -hgd switch for the browser to locate the graphs. The following command should all be on one line.

```
analyze -dtoo -ptoo -wide -png -t -html -pd /var/skippy/ps -gd /var/skippy/images -hgd ../images  
/var/skippy/procstat/20110505 > /var/skippy/ouptut/20110505.html
```

Example B (Linux):

Another example is to use the /opt/sarcheck/etc/sarcheck_parms file. The sarcheck_parms file is a flat file that will change the default behavior of the analyze program. You can include specific keywords that will shorten the analyze command line.

```
PSELFDIR /var/skippy/ps  
GRAPHDIR /var/skippy/images  
WIDE  
HTMLGRAPHDIR ../images
```

```
analyze -png -dtoo -ptoo -t -html /var/skippy/procstat/20110505 > /var/skippy/output/20110505.html
```

How to move SarCheck to another directory on Unix systems

SarCheck by default is located in the /opt/sarcheck directory. There are two ways that you can move SarCheck to another directory. We will give you an example of each way and will use /tmp/sarcheck as the new location of SarCheck.

Example A will use an environment variable called SARCHECKDIR.

Step 1: Edit the file /etc/environment and add the SARCHECKDIR variable
`SARCHECKDIR=/tmp/sarcheck`

Step 2: Create the following directories for SarCheck. /tmp/sarcheck/bin, /tmp/sarcheck/etc, /tmp/sarcheck/ps, /tmp/sarcheck/doc

Step 3: Move the existing files to the new directories
`mv /opt/sarcheck/bin/* /tmp/sarcheck/bin`
`mv /opt/sarcheck/etc/* /tmp/sarcheck/etc`
`mv /opt/sarcheck/ps/* /tmp/sarcheck/ps`
`mv /opt/sarcheck/opt/* /tmp/sarcheck/opt`

Step 4: If SarCheck is running in a process created after this change, it should recognize the environment variable. For details, see the man page for 'environment'.

Step 5: Type `echo $SARCHECKDIR` to verify new location. To eliminate typing you may want to add the new location to your PATH:
`PATH=$PATH:/tmp/sarcheck/bin`

Example B will use the sarcheck_parms file.

Step 1: Create the following directories for SarCheck. /tmp/sarcheck/bin, /tmp/sarcheck/etc, /tmp/sarcheck/ps, /tmp/sarcheck/doc

Step 2: Move the existing files to the new directories
`mv /opt/sarcheck/bin/* /tmp/sarcheck/bin`
`mv /opt/sarcheck/etc/* /tmp/sarcheck/etc`
`mv /opt/sarcheck/ps/* /tmp/sarcheck/ps`
`mv /opt/sarcheck/opt/* /tmp/sarcheck/opt`

Step 3: Create the file /opt/sarcheck/etc/sarcheck_parms (yes, you might have just moved a file with this name) and add the following line to the file with your favorite editor:
`SARCHECKDIR /tmp/sarcheck`

How to access the online instructions / help text

To display the online help text, type:

```
/opt/sarcheck/bin/analyze -h (AIX, Solaris, and Linux)
/opt/sarcheck/bin/analyze9000 -h (HP-UX)
```

This sends a subset of the instructions found in this manual to standard output (stdout), which defaults to the screen.

For other help text type,

```
/opt/sarcheck/bin/analyze -hp or  
/opt/sarcheck/bin/analyze -hm or  
/opt/sarcheck/bin/analyze -hg
```

Note: For HP-UX users replace analyze with analyze9000

A FAQ section can also be found at the end of this manual, and updated information can be found on the SarCheck web site:

<http://www.sarcheck.com/>

How to interpret the analysis

At the beginning of the analysis, the name of the data sar report or Linux procstat data file, the date, time, number of intervals, number of processors seen, amount of memory, and system name is printed for identification purposes.

A warning message will display if the ps data file is not included in the analysis or cannot be found. There are numerous parameters that are included in determining the health of the system that the agent collects and saves in the ps data file. Warning messages will appear if impossible data is seen in the sar report. Examples would be CPU utilization reported by sar -u as 313% or a swap queue occupancy of -88%. The type of sar data or Linux procstat data which contains the problem will be identified. SarCheck will still produce a report, but you should realize that the analysis of anomalous data is, as always, likely to follow the rule of 'garbage in, garbage out'.

The Summary section will highlight any bottlenecks that were seen in the areas of CPU, memory, or I/O, and will indicate if any kernel parameters need to be changed. If no bottlenecks are seen, the summary will say so, and point out that no recommendations will be made.

If runaway processes, memory leaks, or suspiciously large processes have been detected, a message will appear at the end of the Summary section.

The Recommendations section is present only if SarCheck has recommendations to make. If SarCheck thinks that everything is fine, no recommendations will be made. This is a normal condition and once the system is properly tuned, you should not be surprised to see a lack of recommendations.

The recommendations are based solely on the data contained in the sar file or Linux procstat data file and the values of various tunable parameters, and should be taken in that context. For example, if batch jobs are run on Saturdays, and SarCheck analyzes statistics from that day, it may decide that an I/O bottleneck existed and spare memory was present, and therefore, an increase in buffer size may be appropriate. Following these recommendations may improve performance on Saturdays, but could hurt performance during the week by reducing the amount of memory available to users.

The changes to tunable parameters recommended by SarCheck are designed to cause slow, gradual improvement in order to prevent surprises. These gradual changes are designed to prevent any unanticipated side effects of a major change in a tunable parameter.

Due to the interrelationships between tunable parameters and system resources, sarcheck goes beyond the basic rules of thumb whenever possible.

The Resource Analysis section translates the data contained in the sar report or Linux procstat data file into English. Much of this data is provided for reference, and explanations are given where appropriate. The implications of various statistics regarding CPU utilization, buffer sizing, memory utilization, system table sizes, and disk I/O bandwidth are presented in this section.

The times when key resources are most heavily used appear in this section. If these times correlate well with the times that performance degradation was reported, it can be inferred that exhaustion of these resources may be a cause of performance

problems. Peak usage statistics are also used by the capacity planning section.

The Capacity Planning section can be used to approximate the amount of capacity left on the system, based solely on the data being analyzed. CPU, memory, and disk statistics are examined in order to determine which resource is likely to become exhausted first.

This section is not meant to perform the same functions as the more expensive tools available for large systems. It is designed to help meet the needs of system administrators, many of which are managing growing systems and need to know how much "room" is left before various resources become exhausted.

The exhaustion of resources is defined as any single interval in which CPU usage exceeded 90 percent, a disk was busy more than 75 percent of the time, or swapping was detected. *Because the interval with the greatest resource usage is used, the capacity planning report will be less accurate if peak resource use occurred during an interval of less than 10 minutes.* Note that these thresholds can be overridden with the sarcheck_parms file.

The Custom Settings section is where both successful and unsuccessful changes to SarCheck's default thresholds are reported. See the sections "How to change the menu defaults" and "How to change SarCheck's algorithms" for more information.

Disclaimers, trademark information, etc. At the end of the report is a disclaimer, trademark and copyright information, your software serial number, code version, licensee, and if applicable, the software's expiration date.

How to get technical support for SarCheck

Please read the FAQ section of this manual and visit the FAQ section of our website first. If that doesn't do it:

Use our email address: support@sarcheck.com,
call us at +1-207-602-6336, visit our web site at
<http://www.sarcheck.com/>, write to us
at PO Box 1904, Biddeford, ME 04005, USA,
or contact the party from whom you purchased SarCheck.

How to get technical support for gnuplot

See the FAQ at <http://www.gnuplot.info/faq/>

Files included in AIX SarCheck Version 7

/opt/sarcheck/bin/analyze: This program performs the analysis.

/opt/sarcheck/bin/sarcheck: This is the front end for analyze. It's a simple Bourne shell script which allows you to analyze the previous business day's sar data by pressing the enter key a few times. Create a sarcheck_parms file if you want to customize this script. See the section "How to change the menu defaults" for more information.

/opt/sarcheck/etc/analyze.txt: This file contains the text used to produce the analysis. In general, we recommend that you do not modify this file, because it may leave us unable to support the software. Users outside of the United States may modify the spelling of certain words in the file if they wish. For example, the word 'utilization' can be changed to 'utilisation'. If you would like a non-English version of SarCheck, please call us.

/opt/sarcheck/etc/analyze.key: This file contains the activation key. This file is not meant to be edited directly and tampering with it may permanently disable SarCheck.

/opt/sarcheck/bin/fs1: This is a script that calls the program /opt/sarcheck/bin/fsaixagent.

/opt/sarcheck/bin/fsaixagent: The fsaixagent program collects data from the mount, lsvg, and lslv utilities.

/opt/sarcheck/bin/ps1: This is a script that collects ps -elf data, and is roughly analogous to the sa1 script used by sar.

/opt/sarcheck/bin/ps2: This is a script that cleans up ps -elf data, and is roughly analogous to a subset of the sa2 script used by sar.

/opt/sarcheck/bin/scaixagent: This is program called by the ps1 script. It collects data from a number of other utilities and got its name from the fact that in the beginning, it looked only at vmstat -s. The data is stored in the data files found in /opt/sarcheck/ps. Feel free to run this program from the command line if you want to see the output or measure its resource utilization.

/opt/sarcheck/bin/ondemand: This is a script which can be used to get recommendations that are almost "real time". If your system is slow and you want to collect and analyze data while the system is slow, this script will enable you to do it. We are trying to determine if this script meets your needs and haven't received much feedback. Please let us know what you think.

/opt/sarcheck/etc/aixsar15: A sample sar report.

/opt/sarcheck/etc/aixps0515: A sample ps -elf data file.

/opt/sarcheck/etc/aixfs0515: A sample fs data file.

/opt/sarcheck/doc/readme_aix7: An overview of the new enhancements to Version 7.

/opt/sarcheck/ps/readme: This is a short description of the purpose of the ps directory.

/opt/sarcheck/fs/readme: This is a short description of the purpose of the fs directory.

/opt/sarcheck/doc/scdoc701B.pdf: The documentation for SarCheck.

/opt/sarcheck/doc/aixv7report.pdf: This is an example SarCheck report with explanations of the different sections.

/opt/sarcheck/doc/scoptionsv7B.pdf: The "Options available when running 'analyze'" pdf lists the available switches to be used with the analyze command to customize SarCheck's output.

/opt/sarcheck/doc/scparmsv7B.pdf: The "SarCheck parms file keywords" pdf lists keywords that can be used in the /opt/sarcheck/etc/sarcheck_parms file. The keywords will change SarCheck's default behavior and enable you to customize the output.

/opt/sarcheck/etc/sarcheck_parms: This file is not actually included with the SarCheck distribution but you might want to create it in order to modify the SarCheck menu defaults or the thresholds used by SarCheck's algorithms.

Files included in Solaris SarCheck Version 7

/opt/sarcheck/bin/analyze: This program performs the analysis.

/opt/sarcheck/bin/sarcheck: This is the front end for 'analyze'. It's a simple Bourne shell script which allows you to analyze the previous business day's sar data by pressing the enter key a few times. Feel free to customize this script to meet your needs, but save an unmodified copy in order to be safe.

/opt/sarcheck/etc/analyze.txt: This file contains the text used to produce the analysis. In general, we recommend that you do not modify this file, because it may leave us unable to support the software. Users outside of the United States may modify the spelling of certain words in the file if they wish. For example, the word 'utilization' can be changed to 'utilisation'. If you would like a non-English version of SarCheck, please call us.

/opt/sarcheck/etc/analyze.key: This file contains the activation key. Tampering with this file may permanently disable SarCheck.

/opt/sarcheck/bin/ps1: This is a script that collects data from ps -elf and from the dnldmon program, and is roughly analogous to the sa1 script used by sar.

/opt/sarcheck/bin/ps2: This is a script that cleans up ps -elf data, and is roughly analogous to a subset of the sa2 script used by sar.

/opt/sarcheck/bin/scsolagent: This program collects the values of various kstat & sysconf data.

/opt/sarcheck/bin/solst1: a script that collects data from scsolstatic program.

/opt/sarcheck/bin/solst2: This is a script that cleans up data collected by the scsolstatic program. The data is located in the /opt/sarcheck/static directory.

/opt/sarcheck/bin/scsolstatic: This program saves the disk mapping for disk devices using the c0t0d0 naming conventions as well as the sd334 convention used in the sar report. The data is needed when the ctd switch or the CTD sarcheck_parms keyword is used in the analysis.

/opt/sarcheck/static/readme: This is a short description of the purpose of the static directory

/opt/sarcheck/doc/readme_sol7: An overview of the new enhancements to Version 7.

/opt/sarcheck/ps/readme: This is a short description of the purpose of the ps directory.

/opt/sarcheck/doc/scdoc701B.pdf: The documentation for SarCheck.

/opt/sarcheck/doc/scoptionsv7B.pdf: The "Options available when running 'analyze'" pdf lists the available switches to be used with the analyze command to customize SarCheck's output.

/opt/sarcheck/doc/scparmsv7B.pdf: The "SarCheck parms file keywords" pdf lists keywords that can be used in the /opt/sarcheck/etc/sarcheck_parms file. The keywords will change SarCheck's default behavior and enable you to customize the output.

/opt/sarcheck/etc/solsar14: A sample sar report. Recommendations produced by this test file should not be implemented, since the problems which generate the recommendations do not exist on your system.

/opt/sarcheck/etc/solps14: A sample ps -elf report collected with the /opt/sarcheck/bin/ps1 script. Recommendations produced by this test file should not be implemented, since the problems which generate the recommendations do not exist on your system.

/opt/sarcheck/etc/solsta14: A sample static data file collected by the scsolstatic program.

/opt/sarcheck/bin/ondemand: This is a script which can be used to get recommendations that are almost "real time". If your system is slow and you want to collect and analyze data while the system is slow, this script will enable you to do it.

Files included in HP-UX SarCheck Version 7

/opt/sarcheck/bin/analyze9000: This program performs the analysis.

/opt/sarcheck/bin/schpagent: This program collects data about memory and buffer cache utilization from the kernel. It is run by the /opt/sarcheck/bin/ps1 script and the data is stored with the ps -elf data collected by the same script.

/opt/sarcheck/bin/sarcheck: This is the front end for analyze9000. It's a simple Bourne shell script which allows you to analyze the previous business day's sar data by pressing the enter key a few times. Feel free to customize this script to meet your needs, but save an unmodified copy in order to be safe.

/opt/sarcheck/etc/analyze_txt: This file contains the text used to produce the analysis. In general, we recommend that you do not modify this file, because it may leave us unable to support the software. Users outside of the United States may modify the spelling of certain words in the file if they wish. For example, the word 'utilization' can be changed to 'utilisation'. If you would like a non-English version of SarCheck, please call us.

/opt/sarcheck/etc/analyze_key: This file contains the activation key. Tampering with this file may permanently disable SarCheck.

/opt/sarcheck/bin/ps1: This is a script that collects ps -elf data, and is roughly analogous to the sa1 script used by sar.

/opt/sarcheck/bin/ps2: This is a script that cleans up ps -elf data, and is roughly analogous to a subset of the sa2 script used by sar.

/opt/sarcheck/bin/ondemand: This is a script which can be used to get recommendations that are almost real-time. If your system is slow and you want to collect and analyze data while the system is slow, this script will enable you to do it. This script is new and we are trying to determine if it meets your needs. Please let us know what you think.

/opt/sarcheck/bin/vg1: This is a script that runs the program /opt/sarcheck/bin/vgparse.

/opt/sarcheck/bin/vgparse: This program runs the vgdisplay and pvdisplay utilities and puts the output in the ps -elf used by SarCheck. SarCheck uses this data to report on volume groups and physical volumes.

/opt/sarcheck/doc/readme_hp7: An overview of the new enhancements to Version 7.

/opt/sarcheck/doc/scdoc701B.pdf: The documentation for SarCheck.

/opt/sarcheck/doc/scoptionsv7B.pdf: The "Options available when running 'analyze'" pdf lists the available switches to be used with the analyze command to customize SarCheck's output.

/opt/sarcheck/doc/scparmsv7B.pdf: The "SarCheck parms file keywords" pdf lists keywords that can be used in the /opt/sarcheck/etc/sarcheck_parms file. The keywords will change SarCheck's default behavior and enable you to customize the output.

./hpsar22: A sample sar report.

Files included in Linux SarCheck Version 7

/opt/sarcheck/bin/analyze: This program performs the analysis.

/opt/sarcheck/bin/sarcheck: This is the front end for analyze. It's a simple Bourne shell script which allows you to analyze the current day's data by pressing the enter key a few times. Create a sarcheck_parms file if you want to customize this script. See the section "[How to change the menu defaults](#)" for more information.

/opt/sarcheck/etc/analyze.txt: This file contains the text used to produce the analysis. In general, we recommend that you do not modify this file, because it may leave us unable to support the software. Users outside of the United States may modify the spelling of certain words in the file if they wish. For example, the word 'utilization' can be changed to 'utilisation'. If you would like a non-English version of SarCheck, please call us.

/opt/sarcheck/etc/analyze.key: This file contains the activation key. This file is not meant to be edited directly and tampering with it may permanently disable SarCheck.

/opt/sarcheck/bin/sclinagent: This is the program that collects data from the /proc filesystem and other locations.

/opt/sarcheck/bin/prst1: This is a script that runs the sclinagent program.

/opt/sarcheck/bin/prst2: This is a script that deletes old data created by prst1.

/opt/sarcheck/bin/ps1: This is a script that collects ps -elf data.

/opt/sarcheck/bin/ps2: This is a script that cleans up ps -elf data.

/opt/sarcheck/etc/20050221: A sample file containing data from /proc/stat and other locations.

/opt/sarcheck/etc/sarcheck_parms: This file is not actually included with the SarCheck distribution but you might want to create it in order to modify the SarCheck menu defaults or the thresholds used by SarCheck's algorithms.

/opt/sarcheck/procstat/readme: This is a short description of the purpose of the procstat directory.

/opt/sarcheck/ps/readme: This is a short description of the purpose of the ps directory.

/opt/sarcheck/doc/readme_lin7: An overview of the new enhancements to Version 7.

/opt/sarcheck/doc/scdoc701B.pdf: The documentation for SarCheck.

/opt/sarcheck/doc/scoptionsv7B.pdf: The "Options available when running 'analyze'" pdf lists the available switches to be used with the analyze command to customize SarCheck's output.

/opt/sarcheck/doc/scparmsv7B.pdf: The "SarCheck parms file keywords" pdf lists keywords that can be used in the /opt/sarcheck/etc/sarcheck_parms file. The keywords will change SarCheck's default behavior and enable you to customize the output.

Frequently asked questions (FAQ)

Q: What UNIX /Linux systems does SarCheck run on?

A: As of this writing, we support AIX versions 5.2 and up, SarCheck might work with AIX versions as old as 4.3. One binary works with both 32- and 64-bit versions of AIX. SarCheck is also available for Solaris SPARC 8+, Solaris x86 10+, HP-UX version 11 and most Linux 2.2 through 2.6 kernels. We will be dropping support for the oldest operating systems at some point in the future. Even if your primary platform is AIX or Solaris, we encourage you to try SarCheck on other platforms so that you can see what it does and understand what we want to do to help sys admins.

Q: What do I need to install if sar is not on my system?

A: There are different file sets for AIX, Solaris, and HP-UX. Please refer to <http://www.sarcheck.com/sarhowto.htm> for additional details.

Q. I just upgraded to AIX 6.1. Why is it warning me about modifications to a restricted tunable?

A. Prior to AIX 6.1 there was no such thing as a restricted tunable. Based on information from IBM's formulas and tuning classes, SarCheck made recommendations which were designed to improve your system's performance. If you upgraded to 6.1 and kept the existing tunable values, the operating system will warn you if the values of restricted tunable have been changed. Please check with IBM to see if the non-default values are appropriate to your environment.

When SarCheck wants a restricted tunable changed on AIX 6.1, it will recommend that you check with IBM before implementing the change and it will provide detailed information about why it has recommended the change so that IBM can understand SarCheck's rationale.

Q. Should I implement recommendations that only show up occasionally?

A. Feel free to try, but first implement the regularly occurring recommendations, since those will address the most frequently occurring problems. If SarCheck occasionally recommends increasing the amount of memory, you should certainly try it. On systems with some extra memory, SarCheck will be able to make additional recommendations that could not be made on systems where memory is "tight".

Q. Why do I get the message "WARNING: Bad sar -b data has been detected"?

A. This is apparently caused by a bug in sar. The presence of negative numbers in sar -b statistics is a bug.

Q. I added the SUNWaccu and SUNWaccr packages, but I can't find the sa1 and sa2 scripts in the crontab entries for sys. Now what? (Solaris)

A. See the section “How to collect data for SarCheck to analyze”.

Q. Why do I get the message "sa[0-3][0-9]: No such file or directory"?

A. This message indicates that there are no sar data files for sarcheck to analyze. The most common cause of this message is that there's a problem with the way sar is set up, or possibly the kernel was built at least 7 days ago and the system has not been rebooted since then.

Q. Why do I get the message "sar[0-3][0-9]: No such file or directory"?

A. This message indicates that there are no sar report files for sarcheck to analyze. The most common cause of this message is that there's a problem with the way sar is set up, or possibly the system is shut down at night before the sa2 script can be run by cron.

Q. Every time I make changes based on SarCheck's recommendations, it makes more recommendations. Why doesn't it just figure out the correct values for all the parameters?

A. That's not how real performance tuning works. There are no “correct” values because tuning is a series of compromises between various system resources. Performance tuning involves a certain degree of trial and error, and gradual change is the only way to do it.

Q. When I try to run sarcheck, I get the message “sarcheck: not found”. What's wrong?

A. Check the following:

- Does the sarcheck script really exist?
(look for /opt/sarcheck/bin/sarcheck).
- Is /opt/sarcheck/bin in your PATH variable?
(echo \$PATH)

Q. Why did SarCheck stop producing reports?

A. Usually this is because the software has expired. Run '/opt/sarcheck/bin/analyze' and look for the expiration date at the bottom of the usage text. If you've licensed SarCheck and the expiration date doesn't make sense to you, run 'analyze -s' and send us the output.

Q. How do I collect data over a 24 hour period?

A. **AIX**

The adm's crontab entries to collect sar data are:

```
0 * * * * /usr/lib/sa/sa1 1200 3 &
45 23 * * * 1-5 /usr/lib/sa/sa2 -i 1200 -A &
```

Root's crontab entries to collect ps -elf and fsagent data are:

```
0,20,40 * * * * /opt/sarcheck/bin/ps1
45 23 * * * /opt/sarcheck/bin/ps2
30 12 * * * /opt/sarcheck/bin/fs1
```

Solaris

The sys' crontab entries to collect sar data are:

```
0,20,40 * * * * /usr/lib/sa/sa1
45 23 * * * 1-5 /usr/lib/sa/sa2 -i 1200 -A
```

For versions prior to Solaris 10 add the following entries to root's crontab file and for Solaris 10 releases add the entries to sys' crontab file to collect the ps -elf, sscolagent and sscolstatic data.

```
0,20,40 * * * * /opt/sarcheck/bin/ps1
45 23 * * * /opt/sarcheck/bin/ps2
```

```
5 12 * * * /opt/sarcheck/bin/solst1
45 23 * * * /opt/sarcheck/bin/solst2
```

HP-UX

The root's crontab entries to collect sar , ps -elf and schpagent data are:

```
0,20,40 * * * * /usr/sbin/sa/sa1
45 23 * * 1-5 /usr/sbin/sa/sa2 -i 1200 -A
0,20,40 * * * * /opt/sarcheck/bin/ps1
45 23 * * * /opt/sarcheck/bin/ps2
30 12 * * * /opt/sarcheck/bin/vgl
```

Linux

su as sys (or root if necessary) and add the following lines to that user's crontab file.

```
0,20,40 * * * * /opt/sarcheck/bin/prst1
45 23 * * 1-5 /opt/sarcheck/bin/prst2
0,20,40 * * * * /opt/sarcheck/bin/ps1
45 23 * * * /opt/sarcheck/bin/ps2
```

Q. How do I collect data every 10 minutes from 08:00 to 18:00 on Mon through Friday?

A. AIX

The adm's crontab entries to collect sar data are:

```
0 8-17 * * 1-5 /usr/lib/sa/sa1 600 6 &
0 18-7 * * 1-5 /usr/lib/sa/sa1 &
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 600 -A &
```

Root's crontab entries to collect ps -elf and fsagent data are:

```
0,10,20,30,40,50 8-17 * * 1-5 /opt/sarcheck/bin/ps1
5 18 * * * /opt/sarcheck/bin/ps2
30 12 * * * /opt/sarcheck/bin/fs1
```

Solaris

The sys' crontab entries to collect sar data are:

```
0 18 * * 1-5 /usr/lib/sa/sa1
0,10,20,30,40,50 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 600 -A
```

For versions prior to Solaris 10 add the following entries to root's crontab file and for Solaris 10 releases add the entries to sys' crontab file to collect the ps -elf, sscologent and sscologstatic data.

```
0,10,20,30,40,50 * * * 1-5 /opt/sarcheck/bin/ps1
5 18 * * 1-5 /opt/sarcheck/bin/ps2
5 12 * * * /opt/sarcheck/bin/solst1
45 23 * * * /opt/sarcheck/bin/solst2
```

HP-UX

The root's crontab entries to collect sar , ps -elf and schpagent data are:

```
0 18 * * 1-5 /usr/sbin/sa/sa1
0,10,20,30,40,50 8-17 * * 1-5 /usr/sbin/sa/sa1
5 18 * * 1-5 /usr/sbin/sa/sa2 -s 8:00 -e 18:01 -i 600 -A
0,10,20,30,40,50 * * * 1-5 /opt/sarcheck/bin/ps1
5 18 * * 1-5 /opt/sarcheck/bin/ps2
30 12 * * * /opt/sarcheck/bin/vgl
```

Linux

su as sys (or root if necessary) and add the following lines to that user's crontab file.

```
0,10,20,30,40,50 8-17 * * 1-5 /opt/sarcheck/bin/prst1
0 18 * * 1-5 /opt/sarcheck/bin/prst1
5 18 * * 1-5 /opt/sarcheck/bin/prst2
0,10,20,30,40,50 8-17 * * 1-5 /opt/sarcheck/bin/ps1
5 18 * * 1-5 /opt/sarcheck/bin/ps2
```

Q. What's the deal with these IBM tuning recommendations that use lru_file_repage and lru_poll_interval? (AIX)

- A. These parameters have been around since 2006 and yes, we've got them. If we find the presence of the `lru_*` parameters in `vmo`'s output, we'll make the new recommendations. If your operating system is one of the older ones that predate these parameters, we'll make recommendations that use the older algorithms. Either way, you're getting the tuning recommendations that IBM says are the best.

Q. How do I include the physical and logical volume statistics in my report? (AIX)

- A. First make sure to add the following entry to root's crontab file to collect the statistics at 12:30 pm for Monday thru Friday monitoring:

```
30 12 * * * /opt/sarcheck/bin/fs1
```

Command line example: `analyze -dtoo -ps /var/adm/sa/sarxx`

Sarcheck menu example: `sarcheck -dtoo -ps`

The report will include filesystem and disk device tables as well as paragraphs of text. If the analysis is to take place on a different system you will need to bring this `/opt/sarcheck/fs/` data file so it will be included in the analysis.

Q. Why do I get the message "/dev/ksyms is not a 32-bit kernel namelist"? (Solaris)

- A. This message indicates that a 32-bit version of the `/opt/sarcheck/bin/analyze` program is trying to read a 64-bit kernel. Contact us for a 64-bit version of SarCheck.

Q. Why do I get the message "/opt/sarcheck/bin/analyze syntax error at line 1..."? (Solaris)

- A. This message indicates that you are trying to run the 64-bit version of SarCheck on a 32-bit system. Contact us for a 32-bit version of SarCheck.

Q. Why do I get the message "sar: malloc failed Not enough space"?

- A. This is a sar problem, not a SarCheck problem. The message usually means that cron is running the `sa1` script twice instead of once. For example, if your crontab entries look like this:

```
0 * * * 0-6 /usr/lib/sa/sa1
0,20,40 8-17 * * 1-5 /usr/lib/sa/sa1
```

The `sa1` script gets run twice at 8:00 on weekday mornings, and gets run twice every hour after that until 17:00. This example was taken from an actual support call, and the problem disappeared once the second line was changed to this:

```
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
```

Q. Why do I get the gnuplot message "unknown or ambiguous terminal type" when I try to produce graphs?

- A. The version of gnuplot that is installed does not recognize the PNG or JPEG terminal type. To find out what terminal types are supported by your version of gnuplot, run `gnuplot` and type `set terminal` at the prompt.

Q. Why do I get the gnuplot message "line 1: invalid color spec, must be xRRGGBB " when I try to produce graphs?

- A. The version of gnuplot that is installed does not recognize the color spec in the script produced by SarCheck. To find out what version of gnuplot that is on your system, run `gnuplot` and then type `exit` at the prompt. To correct this create a flatfile called `/opt/sarcheck/etc/sarcheck_parms`. The `sarcheck_parms` file is where you customize SarCheck.

If the installed version of gnuplot is 4.0, the sarcheck_parms file should include the following line:

```
GNUPLOT 4.0
```

This will have SarCheck create the correct scripts for your version of gnuplot.

Q. Why do I get the gnuplot message " line 9: undefined variable: off" when I try to produce graphs?

A. The version of gnuplot that is installed does not recognize the script produced by SarCheck. To find out what version of gnuplot that is on your system, run gnuplot and then type exit at the prompt. To correct this create a flatfile called /opt/sarcheck/etc/sarcheck_parms. The sarcheck_parms file is where you customize SarCheck.

If the installed version of gnuplot is 4.2, the sarcheck_parms file should include the following line:

```
GNUPLOT 4.2
```

This will have SarCheck create the correct scripts for your version of gnuplot.

Q. How do I set up SarCheck to analyze weekly reports?

A. To set up SarCheck for weekly reports, you need to concatenate the last 7 days of sar reports into one large file, and then run the analyze program on that file. The easiest way to concatenate the sar reports is probably going to be with the find command. This isn't meant to be a fully debugged and supported solution that's ready to go on a production system, but here is an example of how you would do that:

```
find /var/adm/sa/sar[0-3][0-9] \( ! -mtime +6 \) -exec cat {} \; > /tmp/concat
```

The above command, all on one line, would do the job. Then you'd run analyze on it like this:

```
/opt/sarcheck/bin/analyze -ps /tmp/concat > /tmp/weeklyreport
```

You may want to put these lines in a script and pick a unique name for the weekly report, but this is a generic example of how you'd do it. We don't actually write scripts for people but we can certainly point you in the right direction.

How to deinstall SarCheck

Remove the files mentioned in the section entitled "Files included in this release". By default, they're all located in /opt/sarcheck.