

System Debugging Strategies

By: Micah Altman, (Copyright 1994-7)

Last Revision: Sep. 3, 1997

Monitor the System

By monitoring the network you can catch small problems before they get big, and can help you to catch intermittent problems that may be hard to reproduce.

- *Run periodic tests and benchmarks* to check that system services are working. Keep a record of previous benchmarks to track system performance. However, avoid doing system intensive benchmarks during periods of high system use.
- *Gather performance statistics with tools like sar, fsr, and netcollect* (part of the network visualizer package). Performance and error statistics can point out problems or bottlenecks in particular subsystems and at particular times.
- *Know what is running on your system.* It is often useful to combine statistical information from `sar` with process accounting and auditing information from `satcollect` and `acct`. If necessary, install auditing (`oe.sw.audit`) and accounting (`oe.sw.acct`) software on your system.
- *Keep good records* of setup, performance and changes. Knowing what has changed on a network can speed the troubleshooting process by pointing out areas to investigate.
- *Check log files*, many network problems can be quickly diagnosed by a thorough reading of the log files. A good place to start looking for log messages is the `console` device and the `SYSLOG` file.
 - Remember that `syslogd` can be configured to give different levels of debugging.
 - Remember that some services (like X-windows and DNS) keep additional log files which may contain useful information.

- Check man pages, administrator's guides and release notes the location of log files and how to interpret them.
- *Get more detailed information about problem programs or subsystems:*
 - You can get more information on the startup process if you have set `chkconfig verbose on`, before starting.
 - Ask system services to report more information than in the default configuration. Look for flags that cause daemons and services to operate in *debug*, *trace*, or *verbose* mode (usually `-x`, `-t` or `-v`).
 - Trace programs and scripts that are failing unexpectedly. You can often trace shell scripts by adding a `"-x"` or `"-v"` (or both) to their execution line (e.g. change `"#!/bin/sh"` to `"#!/bin/sh -x"`). You can use `par` to trace any program or daemon. In any case, be prepared for a lot of output.
 - Check man pages, administrator's guides and release notes for flags that affect how the programs, scripts and daemons report errors.

Diagnosing Problems

- *Problem analysis* is be a useful place to start. Compare what the network is doing with what it should be doing. It is difficult to debug a network without knowing what it *should be like*, e.g., the configuration of the systems, which machines are setup as NIS (etc.) servers, which machines are gateways, etc. Knowing what the *performance* of the system has been will help you to define nebulous complaints like "The system is 'slow.'".
- *Compare your system to a similar system that works.* Try to figure out what is different about that system by comparing system directories, modification times, installation lists, and configurations files.
 - If a file is missing on the broken system, use `showfiles` or `versions long` on the working system to show from where (what software package) the file came.

- If an executable complains about a missing library, use `odump` (for static dependencies) and `rld` (for dynamic dependencies) to determine upon which libraries that executable depends. Then use `showfiles` on the working machine to see from the library was installed. (Note that you must have installed the development tools to get this to work.)
- *Make sure services are installed, configured and activated* - the most common problem with system services is a simple failure to install them and turn them on. Software installation can be checked with `versions` (use `versions config` and `versions -cm` to check on configuration files for a software package, but be aware that it may not list some optional configuration files), hardware installation can be checked with `hinv`. Many services are configured with files in `/var/config` and activated by the `network` script when the appropriate `chkconfig` flag is set, but always check the documentation.
- *Isolate problems with individual hosts* from general network problems. Some network problems affect the entire network, some are due to misconfiguration on a single host. Tests several hosts to see if they exhibit identical symptoms. Use a binary *divide and conquer* method to check groups of host instead of checking every host sequentially, when possible.
- *Trace through all the steps in executing a problem service*, failure can occur at any step. Failure at one step can cause symptoms to appear much later. Remember two chains of events are important: the chain of events leading to the services initialization, and the chain of events in execution. Check each step along both of these paths.
- *Look for changes in configuration and setup*. Many problems occur when systems are reconfigured. Check both file modification times and your logs to see what changes occurred in host and system configuration around the time the problem appeared.

Compare the configuration files on a host exhibiting problems to configuration file you know to be correct when possible. When altering configuration files, save the old files, or your changes to these files. Consider using a versioning system such as `scs`.