

## Chapter 43 Basic Troubleshooting - Notes

---

### 43.3 Learning Objectives:

---

- Troubleshoot your system, following a number of steps iteratively until solutions are found.
- Check your network and file integrity for possible issues.
- Resolve problems when there is system boot failure.
- Repair and recover corrupted filesystems.
- Understand how **rescue and recovery** media can be used for troubleshooting.

### 43.4 Troubleshooting Levels

---

There are three levels of troubleshooting:

1. **Beginner:** Can be taught very quickly.
2. **Experienced:** Comes after a few years of practice.
3. **Wizard:** Some people think you have to be born this way, but that is nonsense; all skills can be learned. Every organization should have at least one person at this level of expertise to use as the go-to person.

Even the best administered systems will develop problems. Troubleshooting can isolate whether the problems arise from software or hardware, as well as whether they are local to the system, or come from within the local network or the Internet.

Troubleshooting properly requires judgment and experience, and while it will always be somewhat of an art form, following good methodical procedures can really help isolate the sources of problems in reproducible fashion.

### 43.5 Basic Techniques

---

Troubleshooting involves taking number of steps which need to be repeated iteratively until solutions are found. Basic recipe:

1. Characterize the problem
2. Reproduce the problem
3. Always try the easy things first
4. Eliminate possible causes one at a time
5. Change only one thing at a time; if that doesn't fix the problem, change it back
6. Check the system logs ( `/var/log/messages` , `/var/log/secure` , etc.) for further information.

### 43.6 Intuition and Experience

---

Sometimes, ruling philosophy and methodology requires following very established procedure; making leaps based on intuition discouraged. Motivation for using a checklist and uniform procedure is to avoid reliance on a wizard, to ensure any system administrator will be able to eventually solve a problem if they adhere to well known procedures. Otherwise, if the wizard leaves the organization, there is no one skilled enough to solve tough problems.

If, on the other hand, you elect to respect your intuition and check hunches, should make sure you can get sufficient data quickly enough to decide whether or not to continue or abandon an intuitive path, based on whether it looks like it will be productive.

While ignoring intuition can sometimes make solving a problem take longer, troubleshooter's previous track record is critical benchmark for evaluating whether to invest resources this way. In other word's, useful intuition is not magic, it is distilled

experience.

## 43.7 Things to Check: Networking

---

Following items need to be checked when there are issues with networking:

- **IP configuration**

Use **ifconfig** or **ip** to see if the interface is up, and if so, if it is configured.

- **Network Driver**

If the interface can't be brought up, maybe the correct device driver for the network card(s) is not loaded. Check with **lsmod** if the network driver is loaded as a kernel module, or by examining relevant pseudo-files in `/proc` and `/sys` such as `/proc/interrupts` or `/sys/class/net`.

- **Connectivity**

Use **ping** to see if the network is visible, checking for response time and packet loss. **traceroute** can follow packets through the network, while **mtr** can do this in a continuous fashion. Use of these utilities can tell you if the problem is local or on the Internet.

- **Default gateway and routing configuration**

Run **route -n** and see if the routing table makes sense.

- **Hostname resolution**

Run **dig** or **host** on a URL and see if DNS is working properly.

Network problems can be caused either by software or hardware, and can be as simple as the device driver loaded, or is the network cable connected. If the network is up and running but performance is terrible, it really falls under the banner of performance tuning, not troubleshooting. The problems may be external to the machine, or require adjustment of the various networking parameters including buffer sizes, etc.

## 43.8 Things to Check: File Integrity

---

There are a number of ways to check for corrupt configuration files and binaries. Packaging systems have methods of verifying file integrity and checking for changes, as discussed earlier in this course. For RPM-based systems:

```
$ rpm -V some_package
```

checks a single package, while

```
$ rpm -Va
```

checks all packages on the system.

On Debian-based systems, one can do integrity checking with:

```
$ debsums options some_package
```

which will check the checksums on the files in that package. However, not all packages maintain checksums, so this might not be completely useful. One can also take advantage of the `-v` or `--verify` options on recent versions of **dpkg**.

**aide** does **intrusion detection** and is another way to check for changes in files:

```
$ sudo aide --check
```

will run a scan on your files and compare them to the last scan. Will have to maintain the **aide** database after initializing it, of course.

## 43.9 Boot Process Failures

---

If system fails to boot properly or fully, being familiar with what happens at each stage is important in identifying particular sources of problems. Assuming you get through BIOS stage, may reach one of these unfortunate states:

- **No boot loader screen**

Check for GRUB misconfiguration, or a corrupt boot sector. You may have to re-install the boot loader.

- **Kernel fails to load**

If the kernel **panics** during the boot process, it is most likely a misconfigured or corrupt kernel, or incorrect parameters specified on the kernel command line in the GRUB configuration file. If the kernel has booted successfully in the past, it has either been corrupted, or the kernel command line in the GRUB configuration file has been altered in an unproductive way. Depending on which, you can reinstall the kernel, or enter in the interactive GRUB menu at boot and use very minimal command line parameters and try to fix that way. Or, you can try booting into a rescue image as described in the next chapter.

- **Kernel loads, but fails to mount the root filesystem**

The main causes here are:

- i. Misconfigured GRUB configuration file
- ii. Misconfigured `/etc/fstab`
- iii. No support for the root filesystem type either built into the kernel or as a module in the **initramfs** initial ram disk or filesystem.

- **Failure during the init process**

There are many things that can go wrong once **init** starts; look closely at the messages that are displayed before things stop. If things were working before, with a different kernel, this is a big clue. Look out for corrupted filesystems, errors in startup scripts, etc. Try booting into a lower runlevel, such as 3 (no graphics) or 1 (single user mode).

## Filesystem Corruption and Recovery

---

If during the boot process, one or more filesystems fail to mount, **fsck** may be used to attempt repair. However, before doing that, one should check that `/etc/fstab` has not yet been misconfigured or corrupted. Note once again that you could have a problem with a filesystem type the kernel you are running does not understand.

If root filesystem has been mounted you can examine this file, but `/` may have been mounted as read-only, so to edit the file and fix it you can run:

```
$ sudo mount -o remount,rw /
```

---

to remount it with write permission.

If `/etc/fstab` seems to be correct, you can move to **fsck**. First, should try:

```
$ sudo mount -a
```

to try and mount all filesystems. If this does not succeed completely, can try to manually mount the ones with problems. Should first run **fsck** to just examine; afterwards, can run it again to have it try and fix any errors found.

##

[Back to top](#)

---

[Previous Chapter](#) - [Table of Contents](#) - [Next Chapter](#)