Troubleshooting CentOS

A practical guide to troubleshooting the CentOS 7 community-based enterprise server

Jonathan Hobson
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 5 'Troubleshooting Users, Directories, and Files'
- A synopsis of the book’s content
- More information on Troubleshooting CentOS
Jonathan Hobson is a professional Dev/Ops engineer who provides round-the-clock application and server support to one of the world's largest online newspapers. He has been using CentOS since its inception, and as the author of the best-selling CentOS 6 Linux Server Cookbook, Packt Publishing (ISBN-13: 978-1849519021), Jonathan maintains a strong reputation for the generation of ideas, problem solving, building business confidence, and finding innovative solutions in challenging environments.

Jonathan has worked in a variety of environments, and with more than 20 years of experience as a professional developer, database administrator, and server engineer, he continues to support the open source community at large.

Following a wide range of interests beyond the computer screen, Jonathan also enjoys walking his dogs and getting out and about in the great outdoors.
Preface

CentOS (Community Enterprise Operating System) is known as a robust, stable, and generally trouble-free operating system that is particularly well-suited to the role of a server. As a faithful adaption of RHEL, CentOS has been with us since its initial release in May 2004. It is used by a significant number of servers across the world, by an increasing number of individuals and businesses for a variety of needs, and it can be found in many mission critical situations. CentOS is considered to be a favorite among Linux professionals, and if it is configured correctly, serviced, and maintained, in most instances, a CentOS-based server should never give rise to any major complications. However, there are occasions when things do go wrong, and in such a situation, where the old joke of "rebooting the machine" is not the most appropriate form of action, then your only recourse is to consider troubleshooting the system.

Based on the overall theme of troubleshooting a CentOS 7 server, the purpose of this book is to take you on a journey across the whole spectrum of issue-based problem solving. Active processes, the networking environment, package management, users, directories and files, shared resources, security, databases, web-based services, and DNS will all be encountered with the sole purpose of building your knowledge base and enabling you to develop a fresh approach to problem solving.

What this book covers

Chapter 1, Basics of Troubleshooting CentOS, serves as an introduction to the book, in general, by giving you the low-down on gathering hardware information, dmesg, working with log files, and learning how to manipulate these log files using an array of command-line tools.
Chapter 2, Troubleshooting Active Processes, takes up the running and dives into the world of tuning server performance, swap, memory management, system load, monitoring disk I/O, a tour of the system, guidance on issuing the kill signal, and running additional performance checks using many more tools associated with the command line.

Chapter 3, Troubleshooting the Network Environment, walks you through the process of diagnosing a variety of issues related to the network environment. ping, dig, host, traceroute, mtr, ss, and tcpdump are just some of the tools that will be discussed when highlighting a whole host of network-related problems.

Chapter 4, Troubleshooting Package Management and System Upgrades, puts yum (Yellowdog Updater, Modified) in the spotlight with the intention of showing you how to manage plugins, add additional repositories, download RPM packages, restore the RPM database, and gather generalized software information.

Chapter 5, Troubleshooting Users, Directories, and Files, takes a stance on on-going maintenance and provides the information you need to prepare you for a variety of issues that the professional troubleshooter may face. From user management to login.defs, utmpdump to general file and directory audits. This chapter also builds on your existing knowledge related to the XFS filesystem and shows you how to recover the lost data with Scalpel.

Chapter 6, Troubleshooting Shared Resources, takes a magnifying glass to NFS on CentOS 7 and shows you how to provide shares, manage exports, and access them via a client workstation while simultaneously approaching the subject of CIFS and autofs to deliver an all round problem solving guide.

Chapter 7, Troubleshooting Security Issues, builds on the momentum and discusses why you need to keep SELinux by showing you how to generate audit reports with aureport. From this point onward, you will discover a comprehensive review on FirewallD and an installation guide for Tripwire so that you can develop your very own intrusion detection system.

Chapter 8, Troubleshooting Database Services, lends a hand to troubleshooters and system administrators alike by taking the key points regarding MariaDB, MySQL, and PostgreSQL to provide a birds eye view of how to deal with a lost root password, database tuning, database metrics, and how to install MySQL server on CentOS 7.
Chapter 9, *Troubleshooting Web Services*, takes a step back from recovery and examines the need to improve a system, website, or web application. Taking you through the art of cURL, you will not only discover how to audit your server and access FTP, but you will also learn how to validate your Akamai headers and manage Varnish with the overall intention to illustrate the fine line between Dev/Ops and troubleshooting.

Chapter 10, *Troubleshooting DNS Services*, completes our journey with an investigation into a variety of domain name service issues. Hostnames, FQDNs, BIND, and iftop are all under the knife as we navigate to a number of issues related to bandwidth, cache flushing, and how to make a DNS health check.
Troubleshooting Users, Directories, and Files

Unlike the previous subjects discussed so far, the process of troubleshooting users, directories, and files can be seen as a continual process that requires constant attention during the life time of the server. It will become an every day event, and for this reason, we will start with the basic principles of user management with the intention to show you how to restore the default file and folder permissions, recover lost files, and take you on a journey through many more associated themes in order to prepare you for a variety of issues that any professional troubleshooter may encounter.

In this chapter, we will:

- Learn how to effectively manage the process of adding, deleting, modifying users, and implementing system-wide changes with login.defs
- Discover how to monitor user activity with utmpdump
- Learn how to reset the root password and initiate root-based logging to achieve improved command-line security audits
- Learn how to recover lost data with Scalpel
- Learn how to restore default permissions and ownership
- Discover more about the XFS filesystem by discovering how to run ongoing repairs and investigate defragmentation
- Learn how to audit directories and files
- Discover how to visualize directories and files
Troubleshooting Users, Directories, and Files

Users

User management is a fundamental skill associated with the need to manage a server, and in this respect, it will inevitably represent a milestone when troubleshooting any system. So, with this in mind, we will quickly analyze the process of managing users in order to dispel any confusion.

Adding users and forcing a password change

You can add a new user (and create a home folder for them) by using the following command:

```
# adduser <username>
```

You can provide the new user with a password like this:

```
# passwd <username>
```

Alternatively, if you would like to force a password reset, thereby implying that a user must reset his/her password, then the following command will suffice:

```
# chage -d 0 <username>
```

In addition, you can null a password for a specific user by typing:

```
# usermod -p "" <username>
```

However, if you would like to grant this new user the ability to use `sudo`, then type:

```
# gpasswd -a <username> wheel
```

Finally, if you would like to know more about a user, using the following command, will disclose their current properties:

```
# id <username>
```

Deleting users

The action of deleting a user account is generally straightforward, but it can involve a number of stages that can be forgotten. Therefore, to avoid any future issues across extensive systems with a large amount of users, prior to deleting a user from the system, the account should be locked in the following way:

```
# passwd -l <username>
```
You will then want to back up the home directory using `tar` before confirming if there are any active processes attributed to this account by typing:

```
# ps aux | grep -i <username>
```

Having done this, you can now proceed to kill any active processes attributed to that account by using the following command:

```
# pkill -u <username>
```

Or, you can remove individual process IDs like this:

```
# kill -9 <pid>
```

By using `pkill`, you are invoking the `SIGTERM` command, which will streamline the task of removing any active process associated with that account. So, at this stage, you should now consider removing any files, print jobs, and re-assign or delete any `cron` jobs associated with that account.

You can do this by typing the following command:

```
# find / -user <username> -print
```

Having done this, you can safely delete a user with:

```
# userdel -r <username>
```

Using the `-r` option will also remove the home directory associated with that account, but if you would like to delete the user, their home directory, and remove any `SELinux` mappings, you should use:

```
# userdel -rZ <username>
```

However, if you encounter any difficulties, then you can always use the force option in the following way:

```
# userdel -rfZ <username>
```

Finally, you will need to consider removing any SSH keys associated with that user. Ensure that `sudo` or `su` is not enabled for that account, and then proceed to work through your applications and services one at a time (including database, e-mail, file sharing, `htaccess`, web directories, `CGI` files, and more) while reassigning new settings to any common accounts that the system may use.
Modifying a user

One of the most useful aspects of user management for a troubleshooter is being able to modify an existing user account. There could be many reasons as to why this task is required, but the best illustration of this skill would begin with changing the default adduser attributes in the following file:

```
# nano /etc/default/useradd
```

From here, you can redefine what shell is used, the default location of the home directories, and whether a default mail spool is set.

For example, you can use this technique to change the default location of the home directories from `/home` to `/home/<companyname>`. However, if you prefer to do this manually (on a case-by-case basis), in order to change the location of the home directory, you need to use the `usermod` command in conjunction with the `-d` option (the path to the new directory) and the `-m` option (to move the contents of the current home directory), like this:

```
# usermod -m -d /path/to/new/home/directory <username>
```

When running the preceding command, it is important to realize that a PID will be displayed on the console if the user is currently using the system and this must be killed before any modifications can be made.

Finally, should the need arise to transfer an existing user to a different group, then this can be achieved by invoking the `-g` option like so:

```
# usermod -g <new_group_name> <username>
```

However, having done this, and just as you would for deleting a user, you must manually change the ownership of any `crontab` files or jobs and complete the process by making any relevant changes to any remaining (related/existing) services as well.

Meet login.defs

When it comes to managing users, an alternative or long-term approach is to consider altering the default settings found in `/etc/login.defs` so that you can alter the behavior of the delete command.

For example, consider you find the following line commented out like this:

```
#USERDEL_CMD /usr/sbin/userdel_local
```
Uncomment this line and it will ensure that all `at/cron/print` jobs are removed. Moreover, you can also use the `login.defs` file to determine the default values assigned to the user mail directory, password encryption method, password expiry period, `userid`, `groupid`, and many more.

**Monitoring user activity with utmpdump**

Keeping track of user activity is one of the most essential skills associated with any Linux administrator. In situations where user management may be the cause of a troubleshooting session, we can make use of `utmpdump`.

User histories are typically stored in the following locations:

- `/var/run/utmp`: The purpose of this binary is to record open sessions. You can review the contents of this file with `utmpdump /var/run/utmp`.
- `/var/run/wtmp`: The purpose of this binary is to record connection histories. You can review the contents of this file with `utmpdump /var/log/wtmp`.
- `/var/log/btmp`: The purpose of this binary is to record failed login attempts. You can review the contents of this file with `utmpdump /var/log/btmp`.

Taking this one step further, you can also review the current history of logged sessions contained within `/var/run/wtmp` by typing:

```
# last
```

You can review the current history of logged sessions contained within `/var/run/btmp` by typing:

```
# lastb
```

However, as a simple review of these files is slightly redundant for our needs, you can read the current status of these files with the following commands:

```
# stat /var/run/utmp
# stat /var/log/wtmp
# stat /var/log/btmp
```

The output of these commands may look similar to this:

```
Modify: 2015-04-26 06:24:02.444728081 -0400
Change: 2015-04-26 06:24:02.444728081 -0400
```
Now, given that binary files cannot be viewed using basic reading commands such as `cat`, `less`, and `more`, rather than simply relying on basic commands such as `last`, `who`, `lastb`, and others, a different approach is to use the `utmpdump` command like this:

```
# utmpdump /path/to(binary
```

So, as we have already mentioned earlier, in the case of wanting to read `/var/run/utmp`, you can use the following command:

```
# utmpdump /var/run/utmp
```

While the remaining files would be accessible with:

```
# utmpdump /var/log/wtmp
# utmpdump /var/log/btmp
```

So, having used all three commands, you will then notice that the output is in a familiar format with the most obvious difference being that the results of `wtmp` are displayed in reverse order as opposed to both `utmp` and `btmp` which are displayed in chronological order.

The results of `utmpdump` are formatted in the following way:

- The first column displays a session identifier; the value 7 is typically associated with a new login event, while the value 8 is associated with a logout event.
- The second column displays a PID.
- The third column can hold a relative variable based on either of the following:
  - `--`, indicating a run-level or system reboot change
  - `bw`, or a bootwait process
  - A numeric or TTY value
  - A character/digit that indicates a PTY value (the pseudo terminal).
- The fourth column can sometimes remain empty or maintain an associated username, runlevel, or reboot value.
- The fifth column (if this information is available), will display the TTY or PTY value.
- The sixth column will display the identity of the remote host. In most local cases, you will only see a runlevel message at most, but for remote access, you will see an IP address or name.
• The seventh column will display the remote host’s IP address, or it will show 0.0.0.0 for local access.
• The eighth, and final column, will indicate the time and date information as to when the record was created.

You should also be aware that columns six and seven will show identical information if no DNS resolution is performed.

So, with the preceding information in mind, with a bit of practice, and using the skills we discovered in the previous chapters, utmpdump can be used to perform a wide range of queries such as displaying general access information like this:

# utmpdump /var/log/wtmp

Further to this, you can use grep to show the details of specific records.

For example, if you wanted to display the records of a particular user from wtmp, you will type:

# utmpdump /var/log/wtmp | grep <username>

Taking this one step further, you can use grep to identify the number of logins from a particular IP address in the following way:

# utmpdump /var/log/wtmp | grep XXX.XXX.XXX.XXX

Or use the following syntax to check how many times root accessed the system:

# utmpdump /var/log/wtmp | grep root

Then use the following command to monitor the number of failed login attempts:

# utmpdump /var/log/btmp

Remember, the output of btmp should be minimal, given that this binary will show a variety of issues related to the use of incorrect passwords being used or attempts to log in with an unknown username. The latter of which is particularly important when a tty1 was shown to be used, as this will indicate that an unknown person had access to a terminal on your machine. Look at it this way, noticing such an important issue may inspire you to run a security audit on access privileges and keys by creating a basic text-based output file with the following command:

# utmpdump /var/log/btmp > btmp-YYYY-MM-DD.txt
Resetting the root password and enhancing logging

With the release of CentOS 7, you may find that the process of resetting the root password has changed. So, in the event that you forget the root password, you will need to follow these important steps.

Boot the computer and press the E key during the kernel screen phase. On the next screen, scroll down the text and look for the following line:

```
root=/dev/mapper/centos-root ro
```

Now, replace the letters ro with the following:

```
ro init=/sysroot/bin/sh
```

It should then look like this:

```
root=/dev/mapper/centos-root rw init=/sysroot/bin/sh
```

When done, press Control + X or Ctrl + X to boot into the single user mode using the bash shell /sysroot/bin/sh.

In the single user mode, type:

```
# chroot /sysroot
```

After the hash sign (#), type:

```
# passwd root
```

Follow the onscreen instructions and proceed to reset the password, but if you do need to update SELINUX use the command touch /.autorelabel before you do anything else.

When you are ready to finish, type the following command to access the machine in the usual way:

```
# exit
```

Now, reboot your system in the usual way:

```
# reboot
```

Well done! You should now be able to gain full access to the system using the new root password. However, if you decide to update the logging for all system commands, simply open the following file in your favorite text editor like this:

```
# nano /etc/bashrc
```
Scroll down to the bottom and add the following line:

```bash
readonly PROMPT_COMMAND='history -a >(logger -t "$USER[$PWD] $SSH_CONNECTION")'
```

Having done this, you will now find that all the SSH-based command-line activity is logged through `/var/log/messages` like this:

```
192.168.1.183 22: last
192.168.1.183 22: cd /var/log
192.168.1.183 22: cat messages
192.168.1.183 22: last
```

## Recovering lost or deleted files with Scalpel

If a file has been accidentally deleted from the system, you can use a small utility called Scalpel to recover it. Scalpel is a faster alternative to Foremost, which was originally developed by the United States Air Force Office of Special Investigations and The Center for Information Systems Security Studies and Research. Today, it is a tool that is generally associated with both digital forensics investigation and file recovery, and you can install it by typing the following command:

```bash
# yum install scalpel
```

You will need the EPEL repository to complete this process (which is discussed in a previous chapter), but when you are ready, simply update the following configuration file to determine what types of files you would like to search for:

```bash
# nano /etc/scalpel.conf
```

Having done this, you should now create a recovery directory, and then you should move to the `/etc` directory in order to use `scalpel.conf` like this:

```bash
# cd /etc
```

You can run a scan on a relevant device by customizing the following command:

```bash
# scalpel /path/to/device -o /path/to/recovery/directory
```
Troubleshooting Users, Directories, and Files

An example of the preceding command would look like this:

```bash
# scalpel /dev/sda1 -o /tmp/recovery-session1
```

Scalpel will begin by creating work queues, but be mindful that the entire operation will take some time to complete. In simple terms, the actual time taken to complete a scan will depend on the disk size, the number of deleted files, the power of the machine in general, and other activities that the system is currently performing.

You can view the findings by using the `ls` command like this:

```bash
# ls -la /path/to/recovery/directory
```

Finally, and before you get started, you should be aware that a new recovery directory must be created every time you run Scalpel (so you may want to consider using an alternative hard disk) as the results will be maintained by a single audit file.

This particular file can be viewed by typing:

```bash
# less /path/to/recovery/directory/audit.txt
```

Remember, Scalpel will work with a variety of filesystem formats or raw partitions, and in this respect, it can be seen as a very useful tool for any troubleshooter.

You can learn more about Scalpel by reviewing the manual like this:

```bash
# man scalpel
```

**Restoring file and directory permissions**

File and directory permissions are important, and to view the current state of all the files in a particular directory, you can run the following command:

```bash
# ll
```

Alternatively, you can target a particular directory by running:

```bash
# ll /path/to/directory
```

However, in a situation where someone has mistakenly changed the permissions of a particular system-based file or folder this calamitous situation can be rectified with the following RPM-based commands:

```bash
# rpm --setugids PACKAGENAME
# rpm --setperms PACKAGENAME
```
On the other hand, should it be the case that an entire directory has been mistakenly updated with `chown` or the `chmod` commands, the following commands will prove more useful:

```bash
# for package in $(rpm -qa); do rpm --setugids $package; done
# for package in $(rpm -qa); do rpm --setperms $package; done
```

Based on the commands shown preceding, the first command will serve to reset all the file and folder ownerships values to the default state, while the second command will serve to reset the relative file permissions. So having run these commands, it is possible that you will see the following messages:

```
chgrp: cannot access '/usr/share/man/zh_TW/man5x': No such file or directory
chown: cannot access '/usr/share/man/zh_TW/man6': No such file or directory
chgrp: cannot access '/usr/share/man/zh_TW/man6': No such file or directory
chown: cannot access '/usr/share/man/zh_TW/man6x': No such file or directory
```

Don't worry! Regardless of which file or directory is listed, such notices can be safely ignored.

### Working with and extending the XFS filesystem

Originally developed at Silicon Graphics in 1993, the main purpose of XFS is to not only support the creation of large filesystems that will allow for metadata journaling, but to provide a technology that can be defragmented and enlarged while mounted and active. This information may or may not be of much use to you as a troubleshooter, but you should be aware that the default filesystem now employed by the most recent release of CentOS is known as XFS. If you did not customize the partitions to any great extent, then you may find that XFS is the filesystem you will be dealing with.

You can quickly confirm the structure of your system with the following command:

```
# df -Th
```
The preceding command (the disk size and partitions ignored) can result in something similar to the following output:

```
Filesystem     Type     Size  Used  Avail   Use% Mounted on
/dev/mapper/centos-root xfs     42G   1.5G   40G    4%  /
devtmpfs        devtmpfs   913M     0  913M     0%   /dev
tmpfs           tmpfs     919M     0  919M     0%   /dev/shm
tmpfs           tmpfs     919M   8.4M  911M    1%   /run
tmpfs           tmpfs     919M     0  919M     0%  /sys/fs/cgroup
/dev/sda1       xfs       494M  139M  356M   29%   /boot
/dev/mapper/centos-home xfs     21G   33M   21G    1%  /home
```

The wording xfs under the column labeled type is what we are looking for. If it is found that your server does use the XFS filesystem, then the XFS tools and utilities file xfsprogs.x86_64 can be installed with the following command:

```
# yum install xfsprogs
```

Generally speaking, you should be aware that XFS can prove to be the source of a subtle loss of performance if the server system is relatively small. In these circumstances ext4 tends to be faster with some single threaded and metadata intensive workloads. Moreover, as shrinking support is not available to XFS, you should know that this technology does not allow the filesystem to be reduced in size even when un-mounted. For this reason, you may want to stay with ext4 when big filesystems or big files are not required.

Looking at the bigger picture, you will be comforted to know that the basic syntax required to create an XFS is similar to other filesystems:

```
# mkfs.xfs /dev/device
```

So, no surprises there, and due to the similarities with other filesystems, I will assume that you are comfortable completing the rest of this procedure. However, before you begin, you should always be aware of the server's hardware configuration before starting this operation, as there may be a few notable issues you may want to be aware of before concluding this operation.

For example, let's say the server exceeded 2 TB. So having completed the initial fdisk operations to build the filesystem layout, (prior to mounting) you may decide to benchmark the system because every good troubleshooter knows that XFS enables write barriers to ensure filesystem integrity.

You can achieve this simple operation by typing:

```
# mount -o inode64 /dev/device /mount/point
```
By default, write barriers will serve to preserve the filesystem from issues relating to power failure, resets, and system crashes, but if your hardware maintains a good write cache, then it may seem more prudent to disable the write barrier in order to reduce the impact on performance.

In this respect, you can mount the device in the following way:

```bash
# mount -o nobarrier /dev/device /mount/point
```

On completion, you can always request further information about a specific volume with the following syntax:

```bash
# xfs_info /mount/point
```

So as we can see, XFS does come with a lot of good features and tools, but when it comes to the process of troubleshooting a server, it is precisely these differences that could be the cause of the problem.

In this respect, and as we will now see, XFS should be treated in a different way to a comparable ext3- or ext4-based system. However, if you need to extend the filesystem, then you will be happy to know that XFS comes complete with a standard tool known as `xfs_growfs` that can be used in the following way:

```bash
# xfs_growfs -d /mount/point
```

Assuming that you have reviewed the man pages, it would be obvious to state that your syntax would use the `-d` option in order to grow the filesystem to the maximum size supported by the device.

---

**Running repairs on XFS**

XFS was created with the intention to support extremely large filesystems. It performs incredibly well under a heavy load and scales with large files, but as a result, it is also susceptible to damage, and it is with this in mind that we now consider a set of tools that will enable us to troubleshoot the server and restore the filesystem.

Known as `xfs_repair`, this tool is used to confirm filesystem consistency and repair any problems that are found. This process will not restore lost data, but it should restore the filesystem on the device in question.

The basic syntax used by `xfs_repair` is as follows:

```bash
# xfs_repair /mount/point
```
However, to avoid any error messages, the procedure will then require that you should initially **umount** the device in question. In this respect, the entire procedure will be as follows:

```
# umount /mount/point
# xfs_repair /mount/point
```

The resulting output will then proceed to run through a series of phases and confirm the relevant events. Once complete, simply remount the device in the usual way to complete the task. However, on the chance that **xfs_repair** fails, repeat this process again but do your research on the respective error messages.

If **xfs_repair** fails to fix the consistency problems on a third occasion, depending on the error messages, you may want to consider an alternative rescue plan for the server, as it should be assumed that data recovery can only be made from backups.

Having said that, it is possible that you can consider additional steps to recover the device in question.

At this current stage, you should assume that data recovery can only be made from backups and your plan is now based on the recovery of the filesystem only. However, having said this, it is important to remember that you should not take any action that will impact the production environment.

It may be possible to restore files from the disk by backing up and restoring the files on the filesystem. To do this, mount the filesystem in the read-only mode and proceed to make a backup with **xfsdump**. From this point onwards, you will want to remake the partition and restore the files with **xfsrestore**. Check **man xfsdump** and **man xfsrestore** for further details.

Alternatively, if log recovery is unsuccessful, it may be possible to recover some of the data by mounting the filesystem in the read-only mode with the **no recover** option. This will avoid running the log recovery process but, by using this method, the filesystem is unlikely to be consistent, and it is to be expected that not all of the data will be returned.

The **xfs_repair** utility is designed to repair filesystems. It is size independent (treating both large and small filesystems equally), but unlike other repair tools, it will not run at boot and it will only initiate logging at mount in order to ensure a consistent filesystem. In cases where **xfs_repair** encounters a damaged log file, it will not be able to repair the filesystem, so in the event that this does happen, you will need to clear the relevant log, mount and then un-mount the XFS filesystem, which is done by adding the **-L** option to force log zeroing like this:

```
# xfs_repair -L /mount/point
```
Remember, resetting the log can leave the filesystem in an inconsistent state. This can, and generally does, result in the loss of data and/or data corruption. So, only apply these methods with the intention to restore the filesystem alone. Remember, the `xfs_repair` command is not intended to restore the data on that filesystem.

**Investigating fragmentation on XFS**

In situations where the filesystem is acting sluggishly, it is possible that fragmentation is impacting your server. In this instance, and if you suspect that fragmentation has occurred or is occurring, then simply run the following command on the relevant device:

```bash
# xfs_db -c frag -r /mount/point
```

By using this command, we are causing `xfs_db` to open the filesystem in a read-only mode (`-r` option) and passing a command (`-c` option) to get the file fragmentation data (`frag`) for the device in question. When we use the `frag` command, it will only return information relevant to the file data in the filesystem as opposed to concerning itself with the fragmentation of free space. So, depending on the specific nature of your system, the resulting output could look similar to this:

```
fragmentation factor 0.31%
```

In a more severe case, it could report the following output:

```
fragmentation factor 93.39%
```

By drawing your attention to the fragmentation factor (expressed as a percentage) in the preceding examples, you may have found at least one reason as to why your server requires troubleshooting. Fixing this situation would be a matter of calling the filesystem organizer utility, otherwise known as `xfs_fsr`. We would simply require the system to reorganize our partition or device to optimize disk usage in a similar way to a Microsoft Windows desktop. In this respect, the most basic syntax for using `xfs_fsr` would be as follows:

```bash
# xfs_fsr /path/to/device
```

Whereas, for a single file, you can use:

```bash
# xfs_fsr /path/to/file
```
However, given that the period of time for these events to complete can be quite long, a more succinct use of this command would be to specify a list of filesystems to reorganize (-m), a time option -t calculated in seconds, and the verbose option -v for a clear indication of what is happening, as follows:

```
# xfs_fsr -m /etc/mtab -t 7200 -v
```

The corresponding output will then display the number of extents that are both before and after the inode. By default, `xfs_fsr` will make ten passes before completing the process unless you decide to reduce the number of passes by using the option -p like this:

```
# xfs_fsr -m /etc/mtab -t 7200 -v -p 2
```

You should be aware that `xfs_fsr` should not be used to defragment the whole system as this is generally regarded to be unnecessary as it can give rise to free space fragmentation, so you can complete this task in stages in the knowledge that the operation can be interrupted cleanly. This will leave the filesystem in a consistent state. If you interrupt the process (using Ctrl + C), `xfs_fsr` will save the defragmentation process to the following location:

```
# /var/tmp/.fsrlast_xfs
```

However, before you dive in, the real issue here is that this fragmentation issue should be approached with caution on a live system, as proceeding to defragment a device or partition during periods of high-load will place an unnecessary burden on your server. So in this instance, the best course of action is to run `xfs_fsr` at a time when the relevant device or partition is not at full load or during lighter working periods.

Finally, and having completed the process of defragmentation, you can confirm the extent of the work performed with the following command:

```
# xfs_db -c frag -r /mount/point
```

So having completed these simple actions, or necessitated a future (and possibly repeat) cron job, you should now notice an immediate improvement with regard to the speed at which files and folders can be moved and transferred.
Auditing directories and files

An important task related to troubleshooting can arise from an understanding of activities commonly associated with the action of reading and writing files. CentOS 7 provides a simple utility for this. Known as auditd, this service (or daemon) starts during the boot process. Events are recorded to an associated log file found at /var/log/audit and as it runs in the background, you can check the current service status with:

```
# systemctl status | grep audit
```

It is possible to customize the auditing service and you can have direct access to manage the log file size, location, and associated attributes by accessing the following file with your favorite text editor:

```
# nano /etc/audit/auditd.conf
```

Moreover, if you do not wish to lose any auditing data, you are able to disable the machine when an audit cannot be performed. To do this, open the configuration file auditd.conf and add or modify the following lines:

```
max_log_file_action = keep_logs
space_left_action = email
action_mail_acct = root
admin_space_left_action = halt
```

This action is severe and it is not something to jump into without doing your homework, but it will serve to remove the default action of rotating log files and replace it with an instruction to e-mail the root user.

Finally, should you wish to take advantage of the audit service flag for every process, simply open /etc/default/grub and add the following argument to the kernel line:

```
audit=1
```

Remember to regenerate grub with the following command and reboot:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

This will ensure that an auditable flag is set for every process after the boot sequence has been initiated and, for even greater simplicity, we can then consider building a unique set of rules by editing the following file:

```
# nano /etc/audit/rules.d/audit.rules
```
To make this as easy as possible, the best approach is to find your server's stig.rules file at /usr/share/doc/audit-X.X.X/stig.rules and copy it to /etc/audit/rules.d/audit.rules. Based on the current package version (in my case), the stig.rules file can be found at /usr/share/doc/audit-2.3.3/stig.rules. Consequently, I ran the following command to create a default rule set:

```
# cp /usr/share/doc/audit-2.3.3/stig.rules /etc/audit/rules.d/audit.rules
```

So, having customized the rules and restarted the auditd service, you will discover that a query can be initiated with the following syntax:

```
# ausearch -f /path/to/directory/or/file
# ausearch -f /path/to/directory/or/file | less
# ausearch -f /path/to/directory/or/file -i | less
```

As an alternative to this, you can use aureport to produce a series of audits in the following way:

To monitor unusual behavior, you can use:

```
# aureport --key --summary
```

To build a report on user logins, you can use:

```
# aureport -l -i -ts yesterday -te today
```

To review access violations, you can try:

```
# ausearch --key access --raw | aureport --file --summary
```

Finally, to review anomalies, you can use:

```
# aureport --anomaly
```

Of course, we haven't covered every aspect of the auditing service, but the preceding examples should get you started. Remember, all of the examples shown can be added to a cron job and, should you wish to know more, the aureport manual can always be viewed any time by typing:

```
# man ausearch
# man aureport
Visualizing directories and files

Good administration starts with good housekeeping, and for this reason, the maintenance of detailed records regarding your server's layout is generally considered to be a good starting point for any Linux administrator. Such a task not only allows you to keep abreast of any changes made to the system as a whole, but it can be a useful approach to debugging. Moreover, because you may have inherited this system, or shared access with a number of administrators, it is probably a good idea to consider running an up-to-date inventory of the changes made.

All directories, folders, and files accessible to a specific Linux-based system are arranged in a single tree. Starting from root (/), this hierarchy may consist of either local or remote files, local or remote filesystem(s), and local or remote block devices.

To view this tree, simply ensure that you have installed the following package:

```bash
# yum install tree
```

By default, the `tree` command will begin indexing from your current location, so to begin, simply change your location to the boot directory like this:

```bash
# cd /boot
```

Now, run the following command:

```bash
# tree
```

The `tree` command is technically described as a recursive directory listing command that displays the content of your server in a tree-like format. It is highly customizable, so if you prefer to target a specific directory from your current location, you can use:

```bash
# tree /path/to/folder
```

You may have noticed that the tree command does not show hidden files by default. Therefore, in order to see all files (including all hidden files), use the `-a` option like this:

```bash
# tree -a /path/to/folder
```

However, if you would like the tree function to restrict itself to displaying folder names only, you should use the `-d` option like this:

```bash
# tree -d /path/to/folder
```

If it all looks a little plain and ordinary, you can add some color to the output with the `-C` option like this:

```bash
# tree -C /path/to/folder
```
Finally, you can combine the preceding options to print the output to a text file by typing:

```
# tree > /folder/name/filename.txt
```

For example, if you wanted to maintain a list of files in one or more directories showing the current permissions, you can use the `-p` option like this:

```
# tree -p > /folder/name/filename.txt
```

Alternatively, if you would prefer to display the output with embedded HTML code for export, try:

```
# tree -H /path/to/folder
```

So, regardless of whether you have adopted a new server, or you are troubled by the number of users accessing and writing files to that server, the `tree` function provides a relative solution to keeping a visual audit of your server, or your devices, by typing:

```
# tree -d /sys/devices
```

So why not combine this with a cron job? Then you can regularly keep an eye on the rise of any potential problems or even maintain a visual record of when those changes took place. In this respect you could assert that the `tree` package is a very useful tool, and to learn more you can review the manual at any time by typing:

```
# man tree
```

**Summary**

In this chapter, we have approached a number of topics related to users, directories, and files, while introducing some related themes associated with the release of the XFS filesystem. From forcing password changes to visualizing the directory structure, restoring the root password to understanding the need for disk defragmentation, our pursuit of troubleshooting CentOS 7 has gone some way to show that the knowledge gained from resolving fundamental system-based issues directly relate to the ongoing human-based issues. It would be true to say that you can never rehearse a disastrous scenario because every event may be unique to one or more systems but, as we have seen, regardless of whether you are monitoring users, modifying users, recovering data, or maintaining the filesystem as a whole, by following a few simple procedures, much of those file, directory, and user-based issues can be solved quickly and efficiently; which leads us gently towards the subject of troubleshooting shared resources.
References

- The Tree Project home page: http://mama.indstate.edu/users/ice/tree/
- XFS FAQ: http://xfs.org/index.php/XFS_FAQ
Where to buy this book

You can buy Troubleshooting CentOS from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.