Tên: Trần Minh Triết

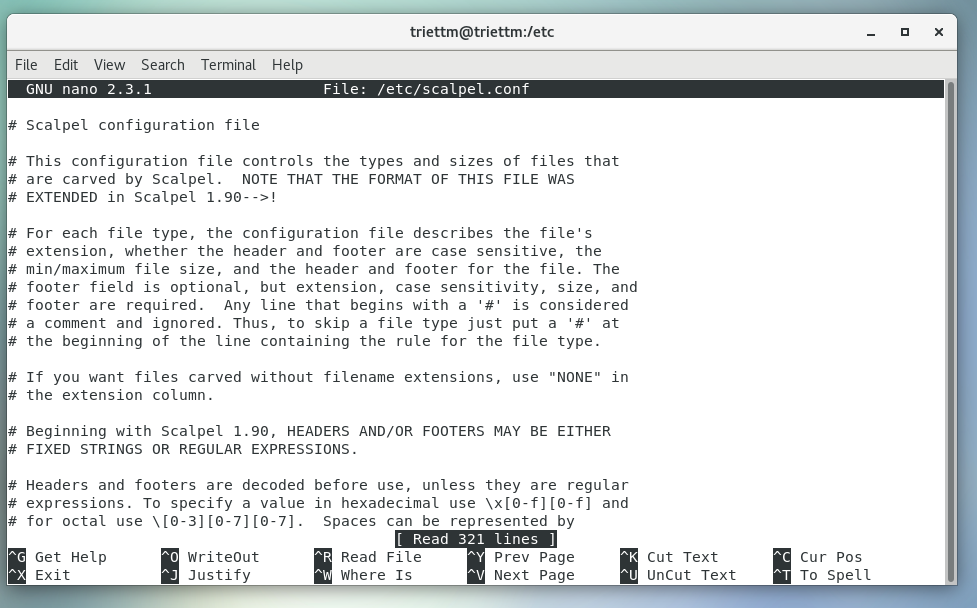
MSSV: SE172241

LAB 6

**Recovering lost or deleted files with 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:

**nano /etc/scalpel.conf**

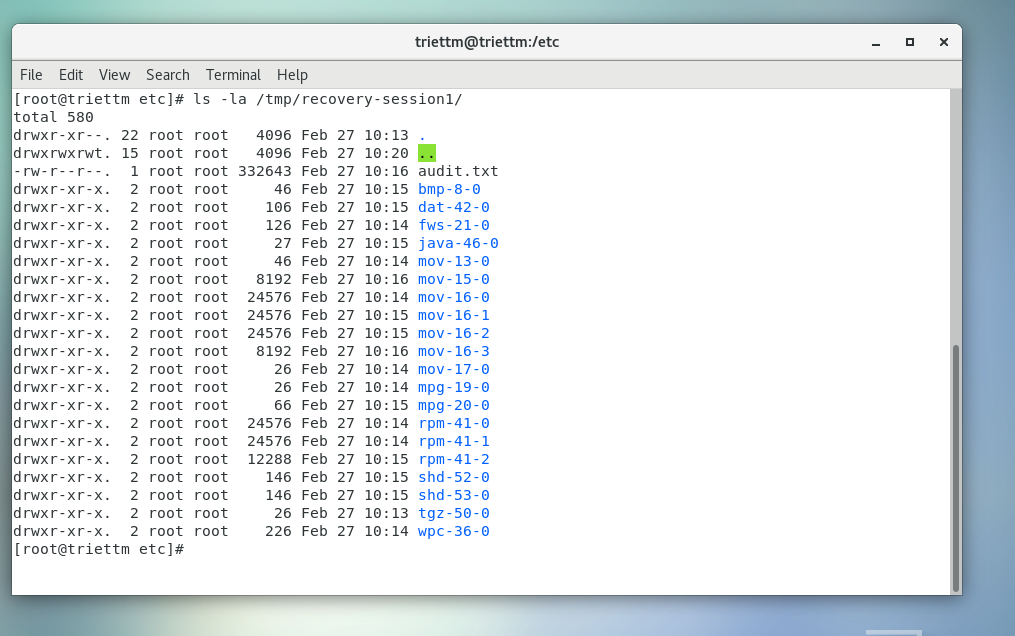


**scalpel /dev/sda1 -o /tmp/recovery-session1**

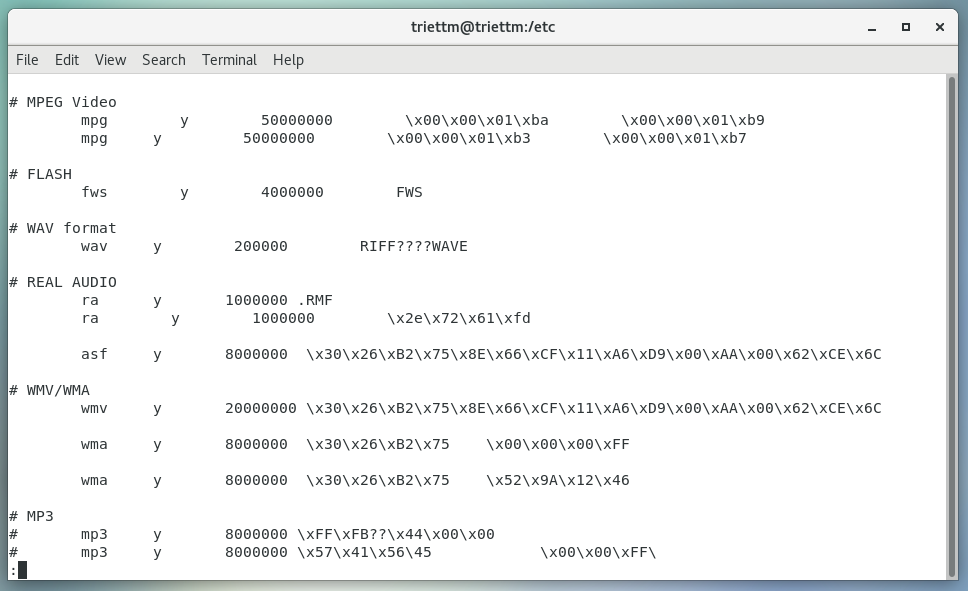
Using the above command, we start using scalpel to recovery data from the disk sda1 to /tmp/recovery-session1

As we do not specify any file type, Scalpel will extract all file types and deleted files to the destination location.

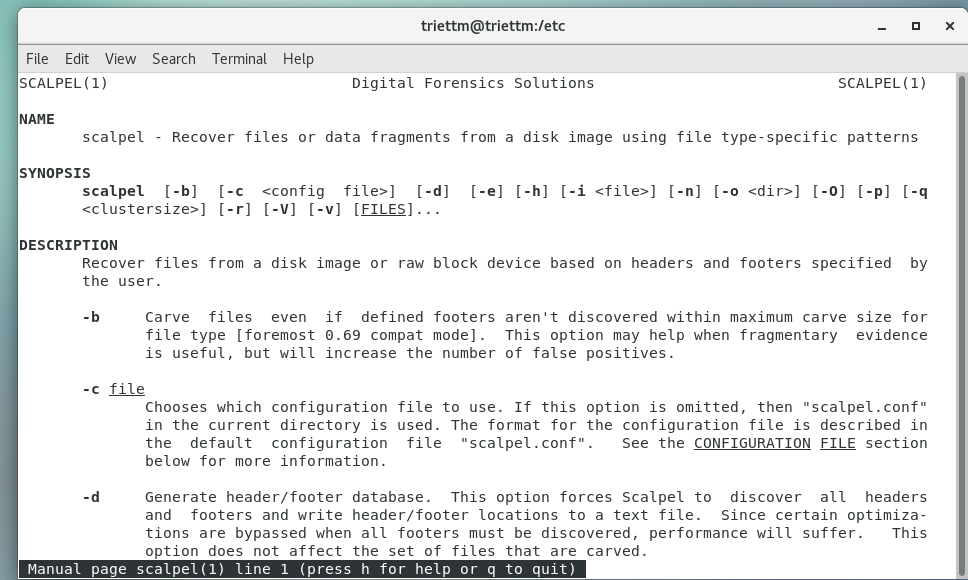
Testing by listing the folder contents



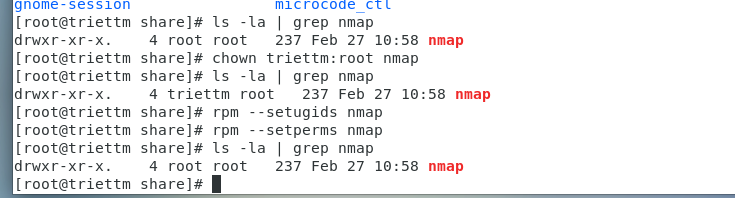
**less /tmp/recovery-session1/audit.txt**



man scalpel



**Restoring file and directory permissions**



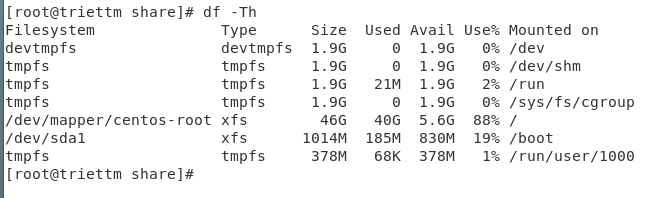
At first I install tool nmap with rpm command:

rpm -vhU <https://nmap.org/dist/nmap-7.93-1.x86_64.rpm>

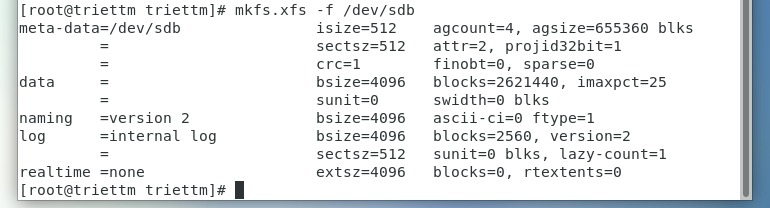
Then as the above image, you can see that the owner of the package is root, group is root.

Then I change the owner of the packet and testing restore its permissions back to root again.

**Working with and extending the XFS filesystem**

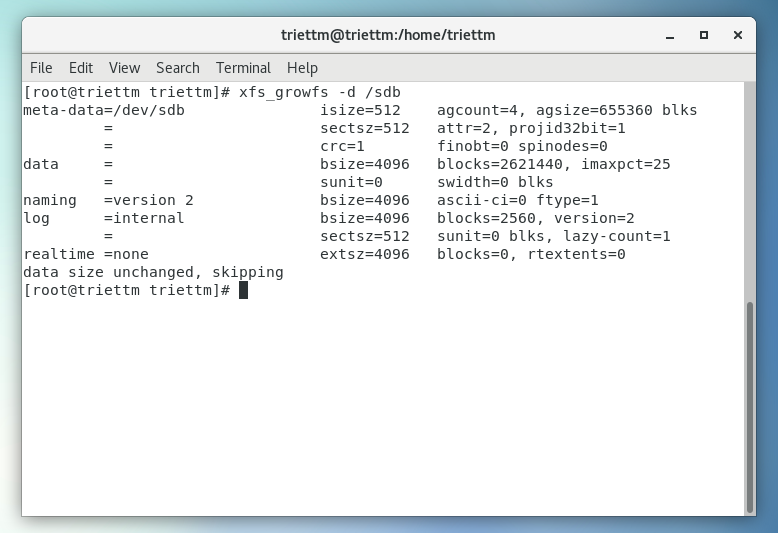


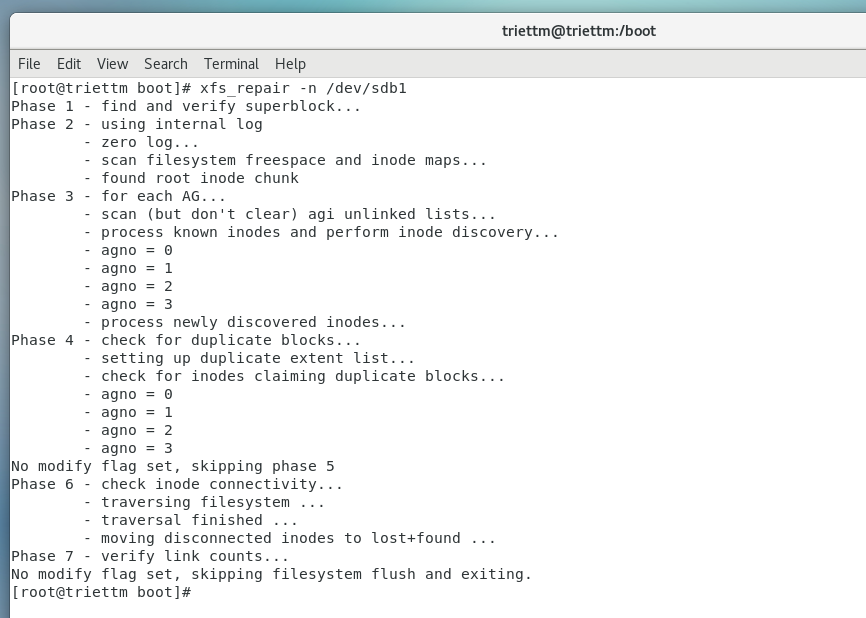
Cấu hình XFS cho ổ cứng

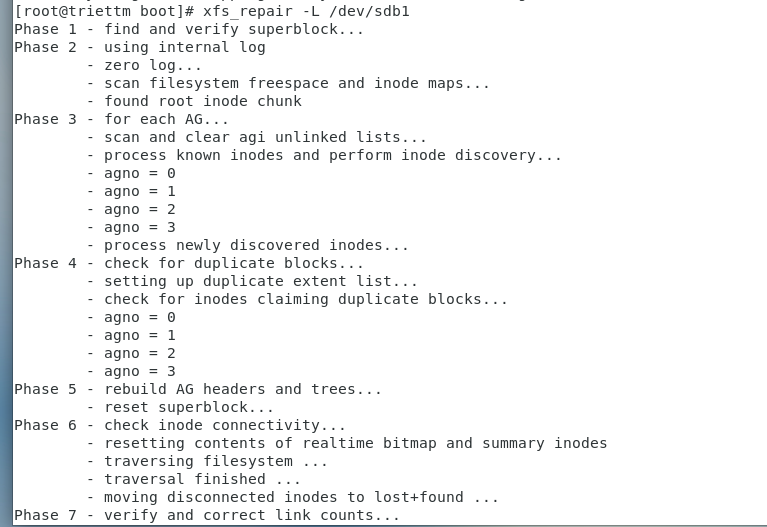


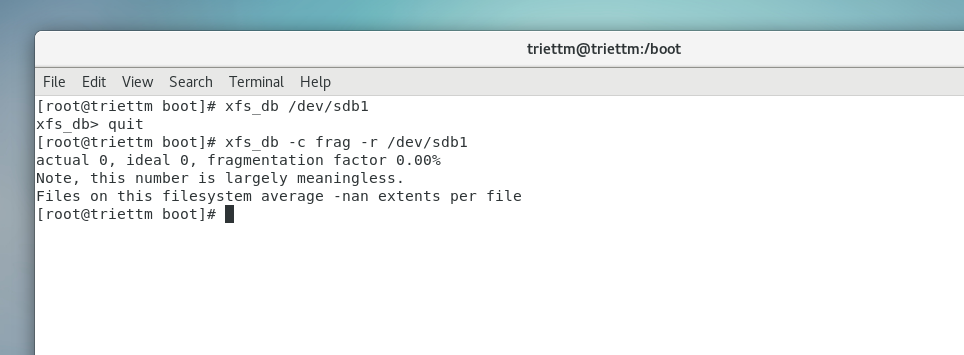
We have successfully config and mount the sdb hard disk with XFS file system.

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:

**Running repairs on XFS**

****

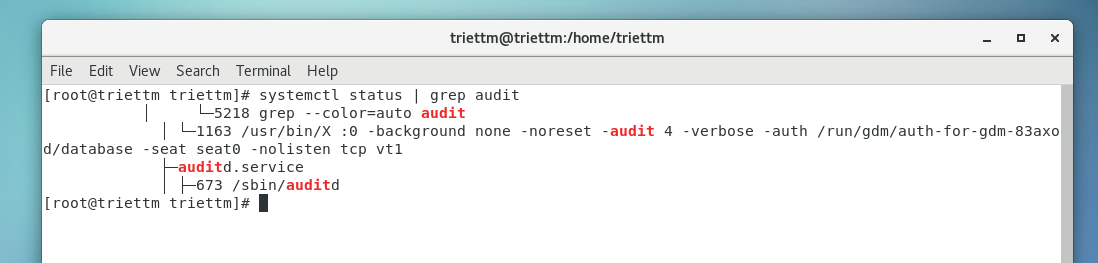
**Investigating fragmentation on XFS**

****

**Auditing directories and files**

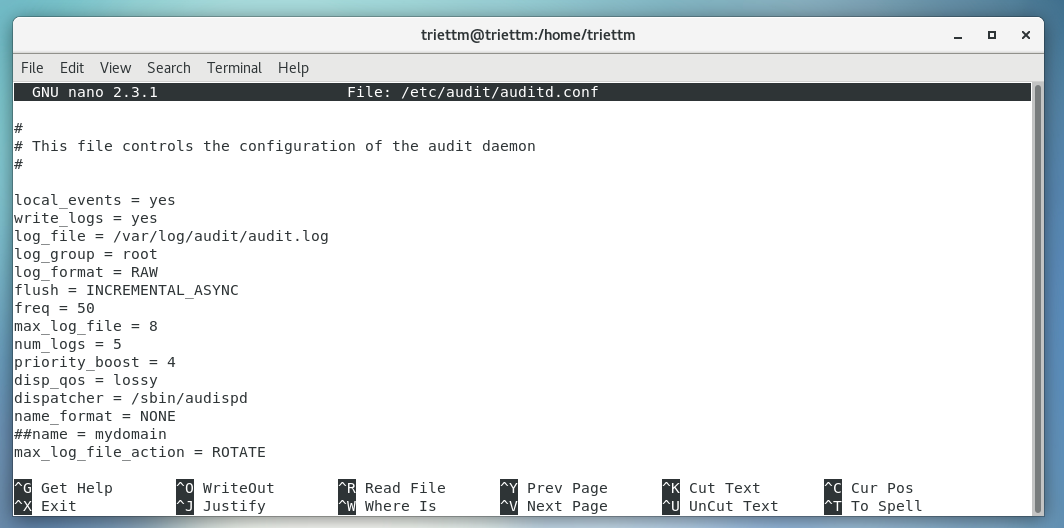
An important task related to troubleshooting can arise from an understanding of activities commonly associated with the action of reading and writing files. CentOS 7 provides a simple utility for this. Known as auditd, this service (or daemon) starts during the boot process. Events are recorded to an 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**

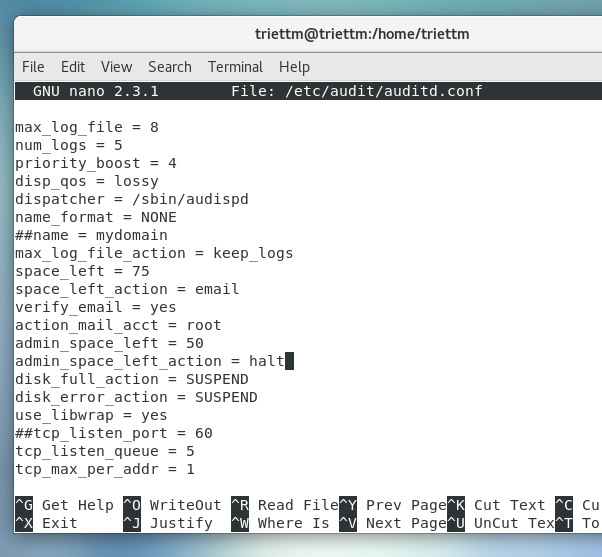


As we can see that the daemon auditd is running in the background.

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:

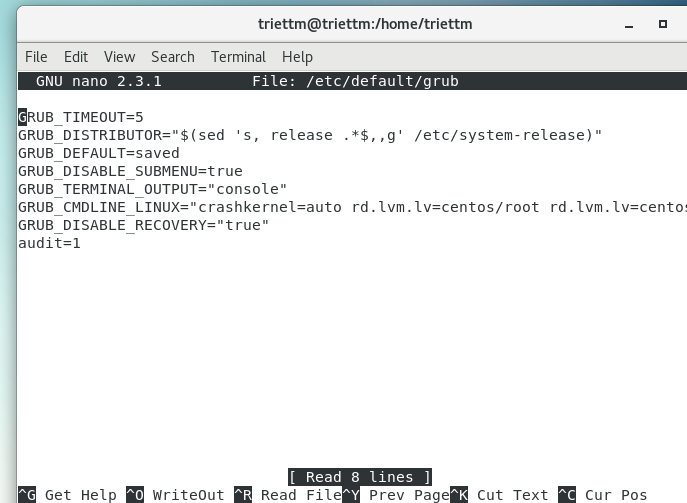


We can change the content of this file to change the behaviour of the auditd daemon.

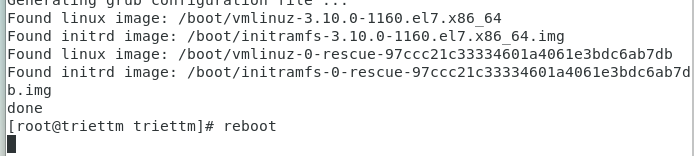


This action is severe and it is not something to jump into without doing your homework, but it will serve to remove the default action of rotating log files and replace it with an instruction to e-mail the root user.

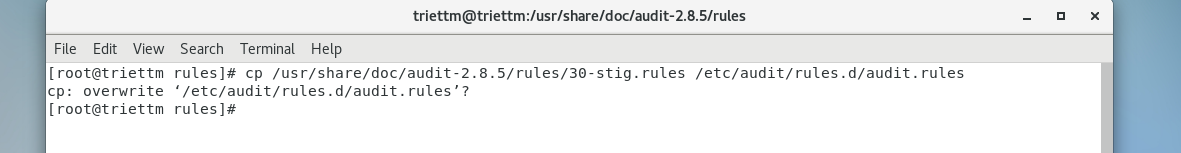
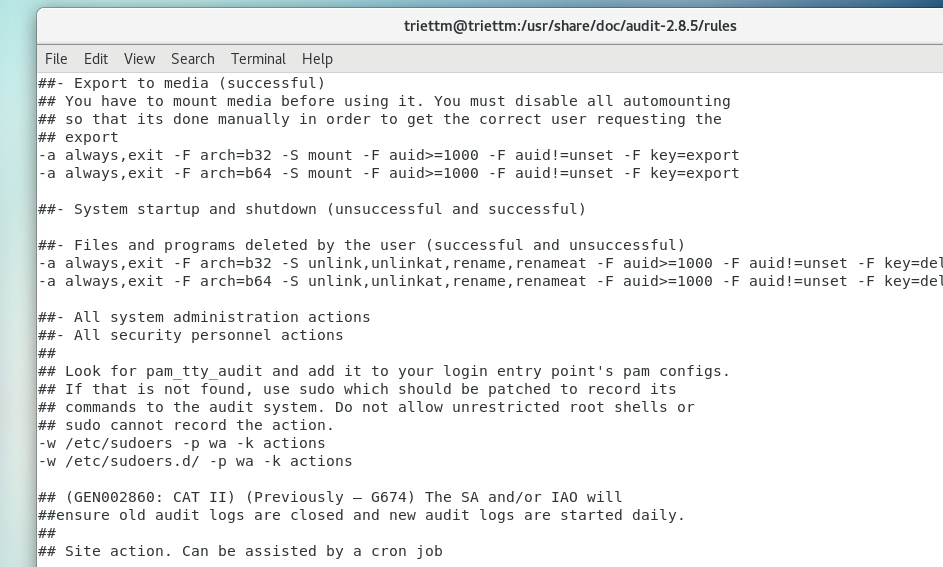
Finally I open the /etc/default/grub to take advantage of the audit service flag for every process.

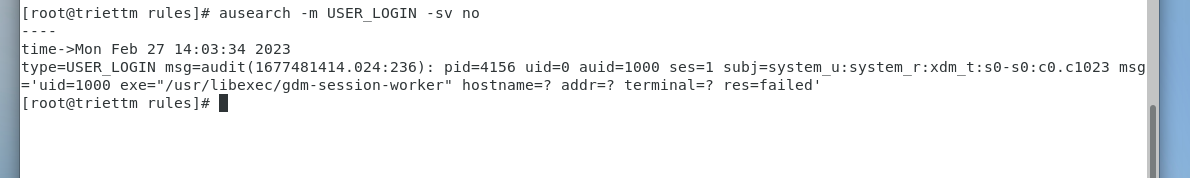


Remember to regenerate grub with the following command and reboot



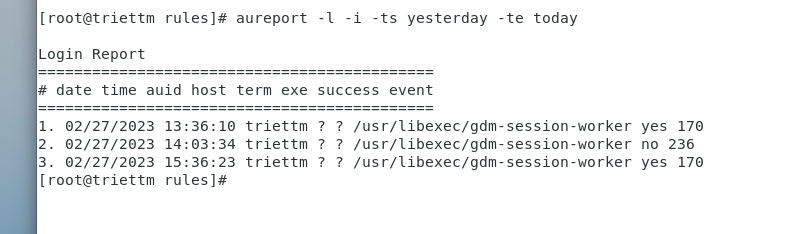
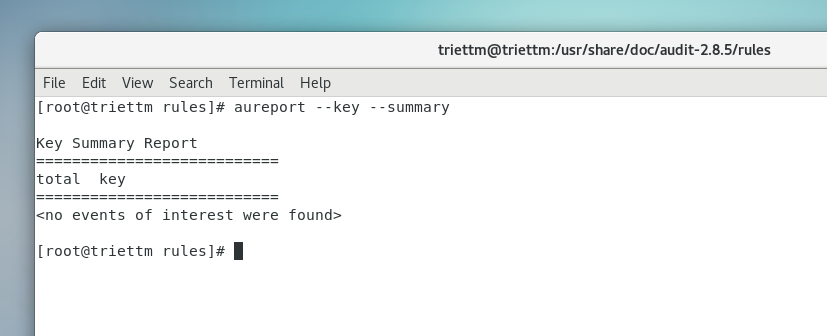
In my computer, the stig.rule file store inside this path /usr/share/doc/audit-2.8.5/rules/30-stig.rules



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

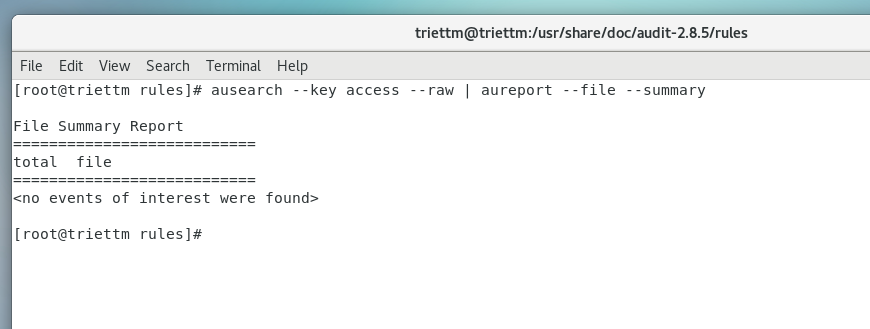
To monitor unusual behavior, you can use:

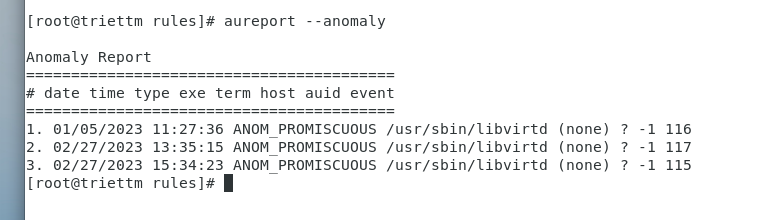
**aureport --key –summary**

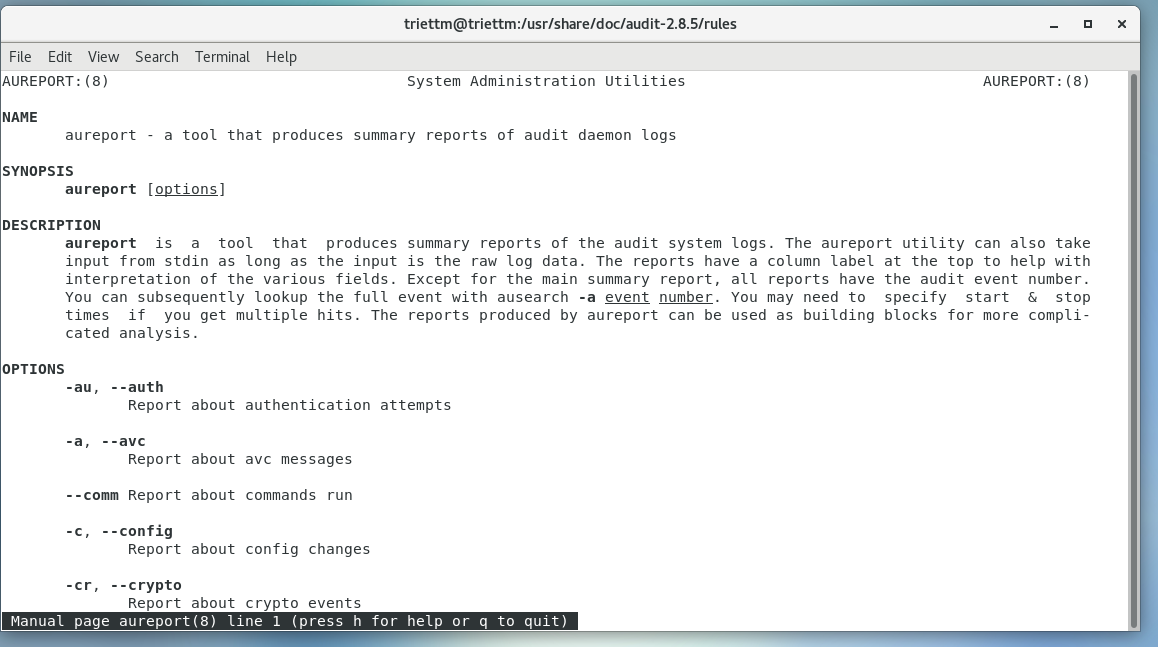
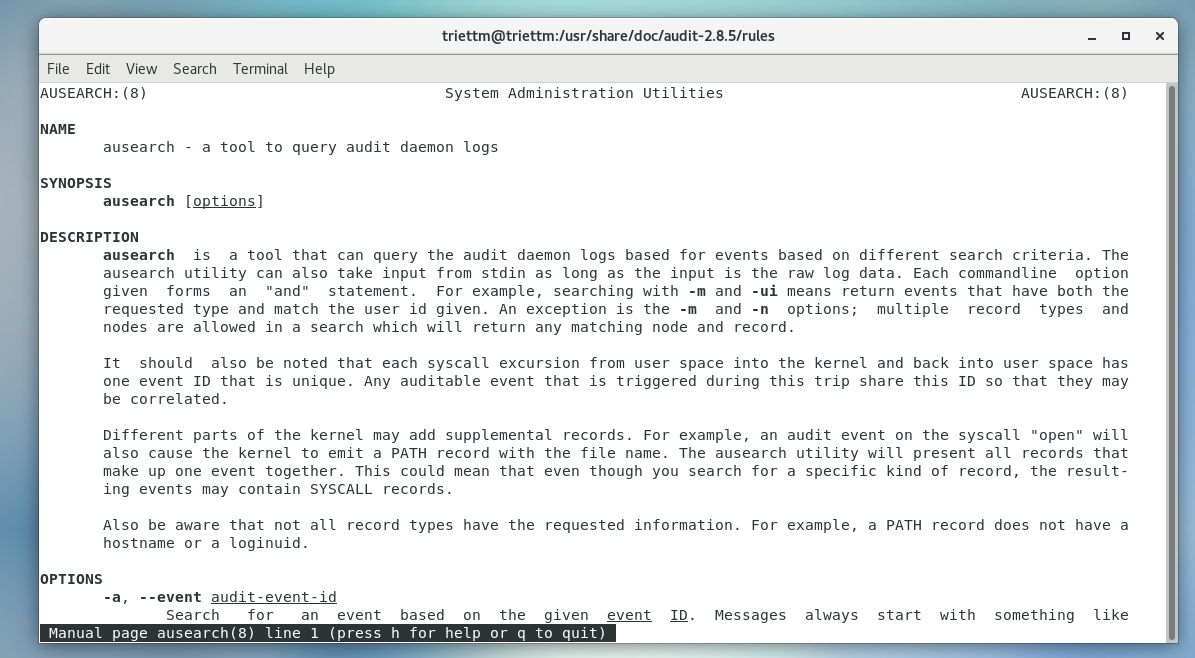


To review access violations, you can try:

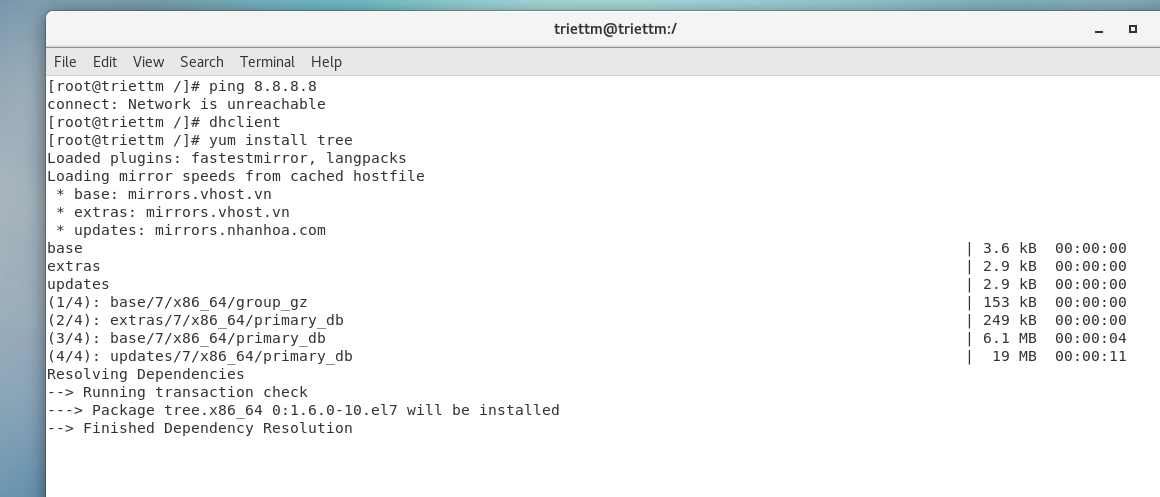
**ausearch --key access --raw | aureport --file –summary**

**aureport –anomaly**





**Visualizing directories and files**



Using yum to install package tree

