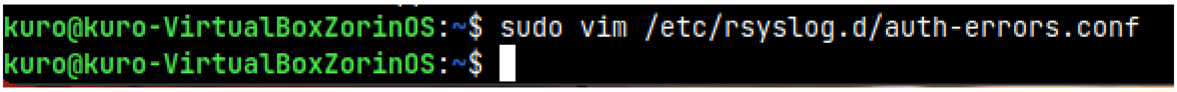
System and Network Administration - Lab 10 - Logging and auditing

```
Jaffar Totanji - j.totanji@innopolis.university
```

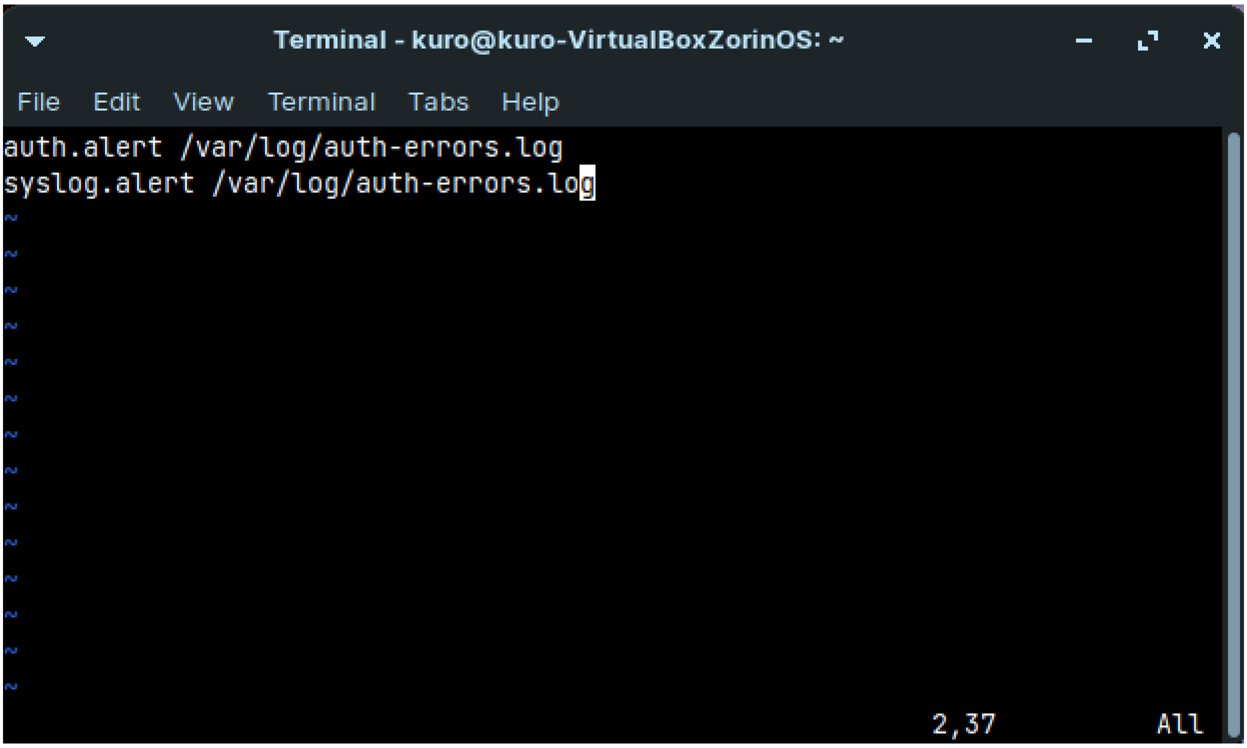
Questions to answer:

1. I would give `Wazuh` (https://wazuh.com/) a shot. Open-source, great product. I also know some great engineers who work there 😃.

2. To do that, we first create a new configuration file `/etc/rsyslog.d/auth-errors.conf`, and populate it with a rule to save all `authentication` and `security` messages with the priority `alert` or higher to `/var/log/auth-errors`:

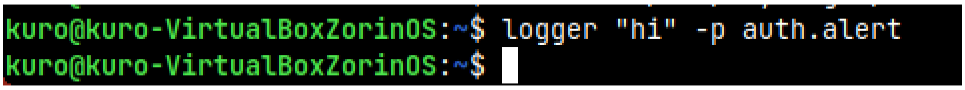


```
kuro@kuro-VirtualBoxZorinOS:~$ sudo vim /etc/rsyslog.d/auth-errors.conf
kuro@kuro-VirtualBoxZorinOS:~$ 
```



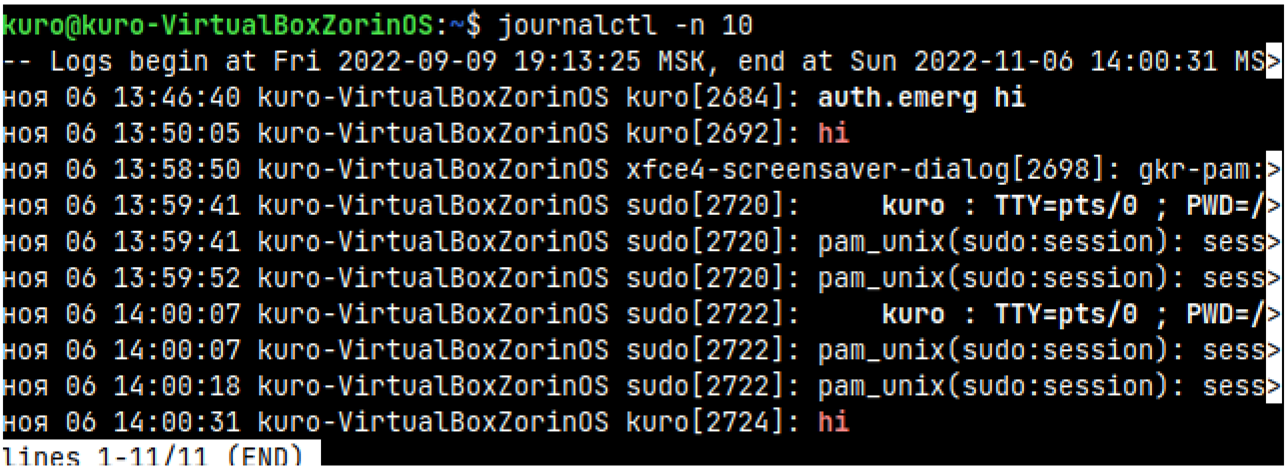
```
auth.alert /var/log/auth-errors.log
syslog.alert /var/log/auth-errors.log
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                  2,37          All
```

We then restart rsyslog using `systemctl restart rsyslog` for the new changes to take effect. To test our new additions, we can log a message of priority `auth.alert` using `logger`:



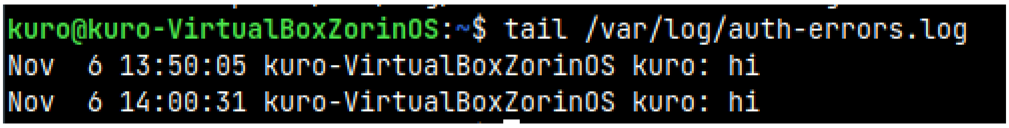
```
kuro@kuro-VirtualBoxZorinOS:~$ logger "hi" -p auth.alert
kuro@kuro-VirtualBoxZorinOS:~$ 
```

We can now see our new log entry using `journalctl`:



```
kuro@kuro-VirtualBoxZorinOS:~$ journalctl -n 10
-- Logs begin at Fri 2022-09-09 19:13:25 MSK, end at Sun 2022-11-06 14:00:31 MS>
ноя 06 13:46:40 kuro-VirtualBoxZorinOS kuro[2684]: auth.emerg hi
ноя 06 13:50:05 kuro-VirtualBoxZorinOS kuro[2692]: hi
ноя 06 13:58:50 kuro-VirtualBoxZorinOS xfce4-screensaver-dialog[2698]: gkr-pam:>
ноя 06 13:59:41 kuro-VirtualBoxZorinOS sudo[2720]:     kuro : TTY=pts/0 ; PWD=/>
ноя 06 13:59:41 kuro-VirtualBoxZorinOS sudo[2720]: pam_unix(sudo:session): sess>
ноя 06 13:59:52 kuro-VirtualBoxZorinOS sudo[2720]: pam_unix(sudo:session): sess>
ноя 06 14:00:07 kuro-VirtualBoxZorinOS sudo[2722]:     kuro : TTY=pts/0 ; PWD=/>
ноя 06 14:00:07 kuro-VirtualBoxZorinOS sudo[2722]: pam_unix(sudo:session): sess>
ноя 06 14:00:18 kuro-VirtualBoxZorinOS sudo[2722]: pam_unix(sudo:session): sess>
ноя 06 14:00:31 kuro-VirtualBoxZorinOS kuro[2724]: hi
lines 1-11/11 (END)
```

and `rsyslog`:

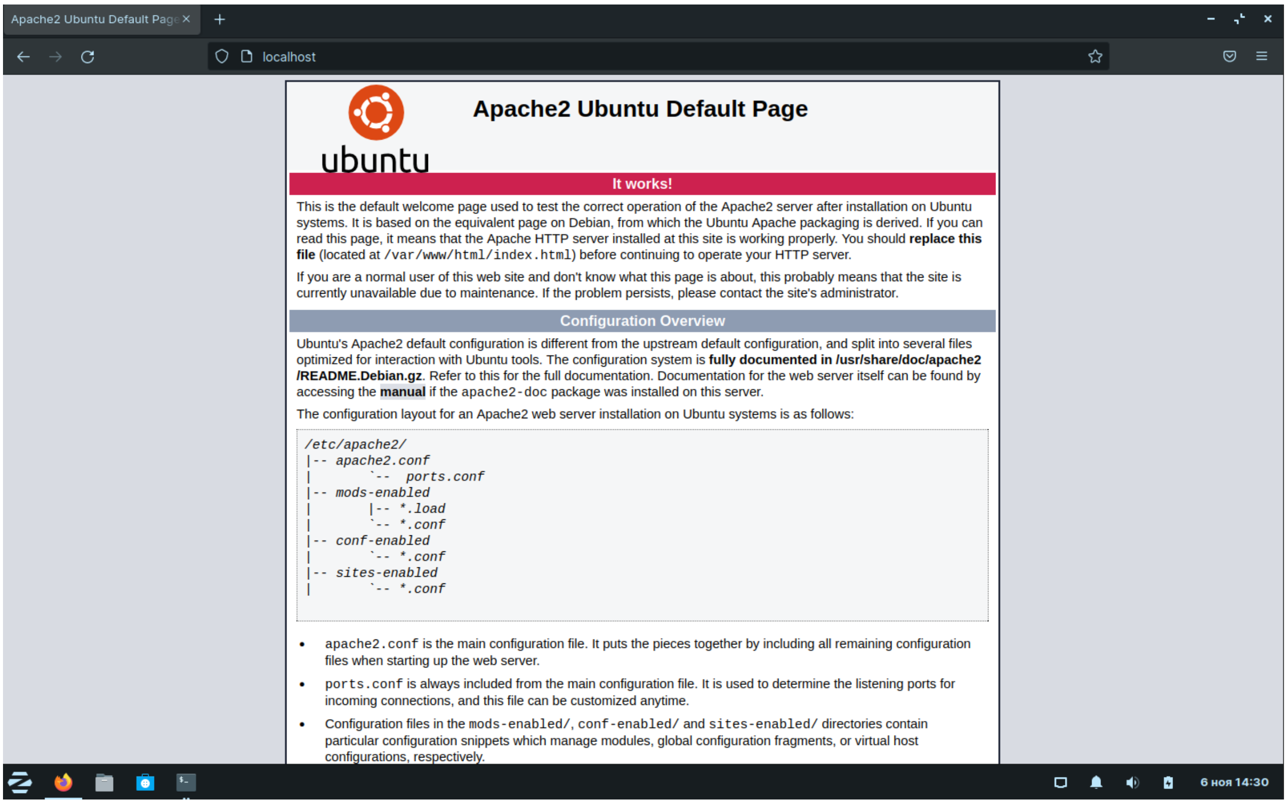


```
kuro@kuro-VirtualBoxZorinOS:~$ tail /var/log/auth-errors.log
Nov  6 13:50:05 kuro-VirtualBoxZorinOS kuro: hi
Nov  6 14:00:31 kuro-VirtualBoxZorinOS kuro: hi
```

3. Let's first install Apache web server:

```
sudo apt update
sudo apt install apache2
```

We can then access `localhost` to verify that it's running:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

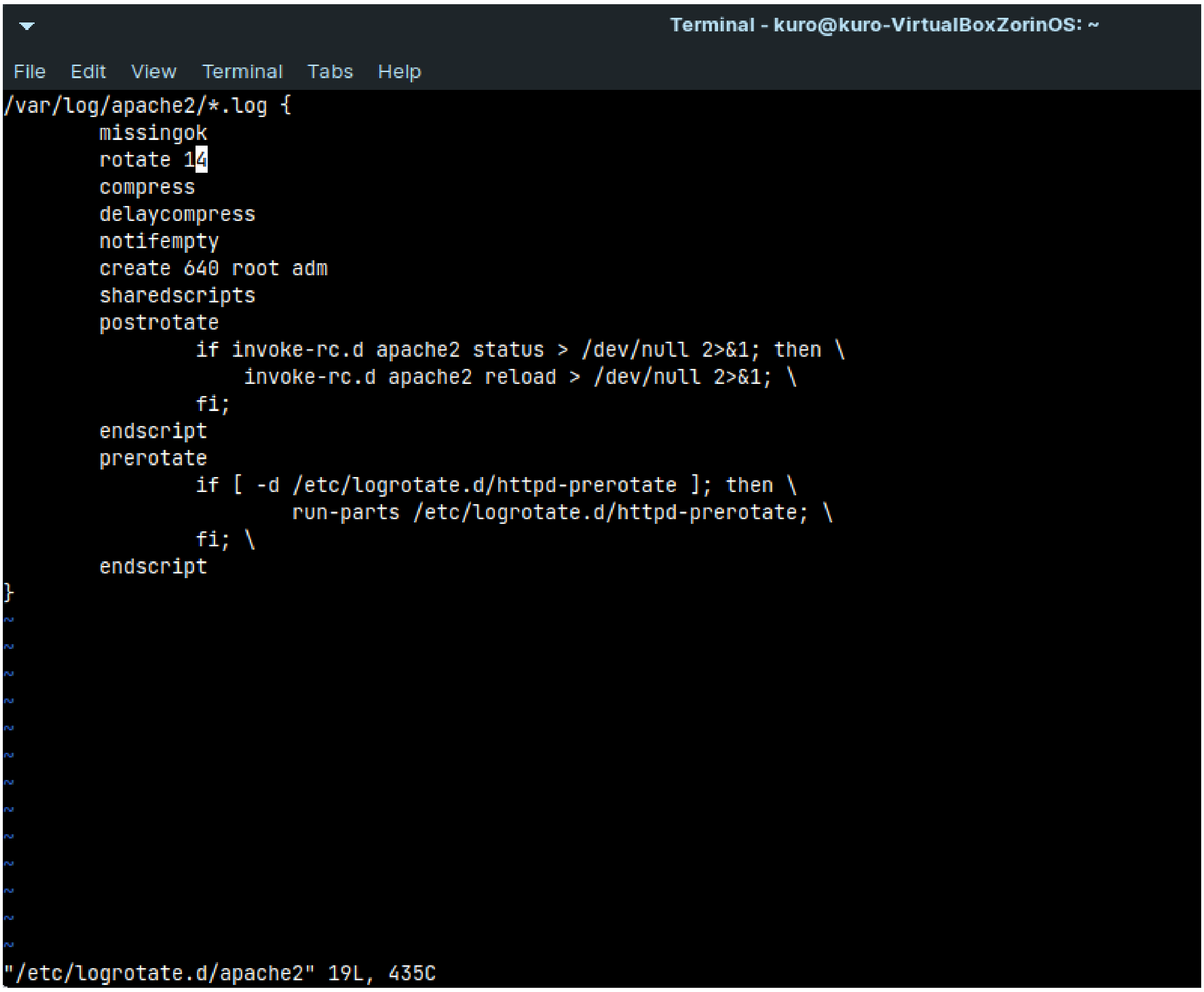
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

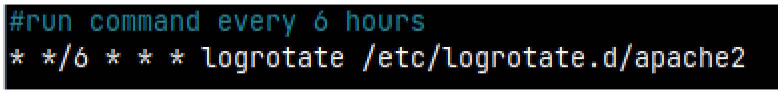
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Apache comes with its own `logrotate` configuration in `/etc/logrotate.d/apache2`, we can edit that to meet our needs. This is what it looks like after removing the default `daily` rotation, it performs various operations including `compression` and `restarting the server`. I have also adjusted it to hold half a week's worth of logs:


Terminal - kuro@kuro-VirtualBoxZorinOS: ~
File Edit View Terminal Tabs Help

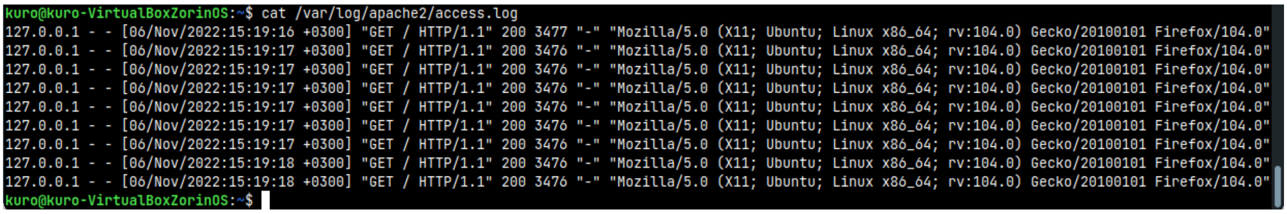
```
/var/log/apache2/*.log {
        missingok
        rotate 14
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                if invoke-rc.d apache2 status > /dev/null 2>&1; then \
                    invoke-rc.d apache2 reload > /dev/null 2>&1; \
                fi;
        endscript
        prerotate
                if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                        run-parts /etc/logrotate.d/httpd-prerotate; \
                fi; \
        endscript
}
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/logrotate.d/apache2" 19L, 435C
```

We can now create a `crontab` to rotate the table every 6 hours:



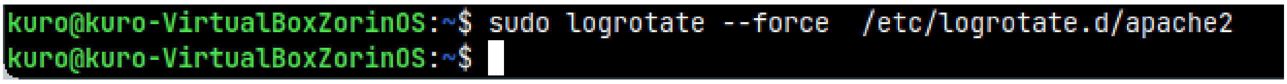
```
#run command every 6 hours
* */6 * * * logrotate /etc/logrotate.d/apache2
```

Let's access the server a couple of times to populate our `access.log`:



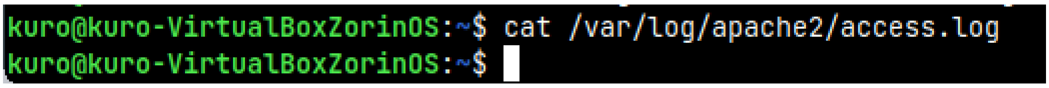
```
kuro@kuro-VirtualBoxZorinOS:~$ cat /var/log/apache2/access.log
127.0.0.1 - - [06/Nov/2022:15:19:16 +0300] "GET / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:17 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:18 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
127.0.0.1 - - [06/Nov/2022:15:19:18 +0300] "GET / HTTP/1.1" 200 3476 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:104.0) Gecko/20100101 Firefox/104.0"
kuro@kuro-VirtualBoxZorinOS:~$
```

We can force the rotation to observe the results instead of waiting 6 hours:



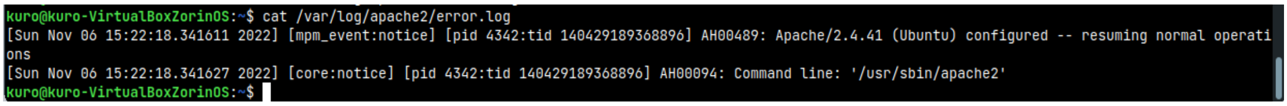
```
kuro@kuro-VirtualBoxZorinOS:~$ sudo logrotate --force  /etc/logrotate.d/apache2
kuro@kuro-VirtualBoxZorinOS:~$
```

Let's take a look at `access.log`:



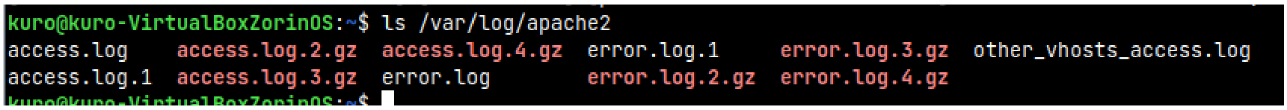
```
kuro@kuro-VirtualBoxZorinOS:~$ cat /var/log/apache2/access.log
kuro@kuro-VirtualBoxZorinOS:~$ 
```

We can tell it has been rotated. And `error.log` can show us that the server has been restarted:



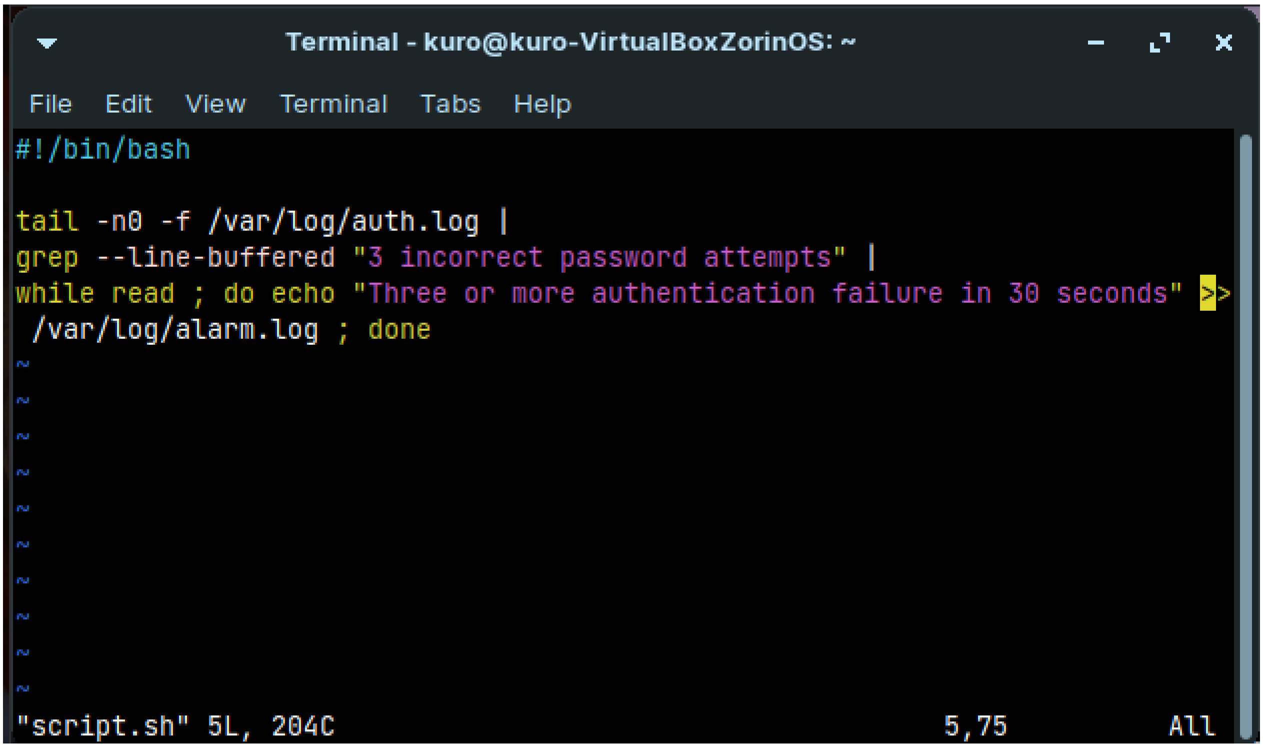
```
kuro@kuro-VirtualBoxZorinOS:~$ cat /var/log/apache2/error.log
[Sun Nov 06 15:22:18.341611 2022] [mpm_event:notice] [pid 4342:tid 140429189368896] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operati
ons
[Sun Nov 06 15:22:18.341627 2022] [core:notice] [pid 4342:tid 140429189368896] AH00094: Command line: '/usr/sbin/apache2'
kuro@kuro-VirtualBoxZorinOS:~$ 
```

We can also see all the old log files:



```
kuro@kuro-VirtualBoxZorinOS:~$ ls /var/log/apache2
access.log     access.log.2.gz  access.log.4.gz  error.log.1     error.log.3.gz  other_vhosts_access.log
access.log.1   access.log.3.gz  error.log        error.log.2.gz  error.log.4.gz
kuro@kuro-VirtualBoxZorinOS:~$ 
```

4. We can create a simple script that continuously checks logs being added to `/var/log/auth.log` and tries to match `3 incorrect password attempts` which is issued by the system upon failing to authenticate a user 3 times within 30 seconds. Upon matching the given text, the script will append the text `Three or more authentication failure in 30 seconds` to `/var/log/alarm.log`:

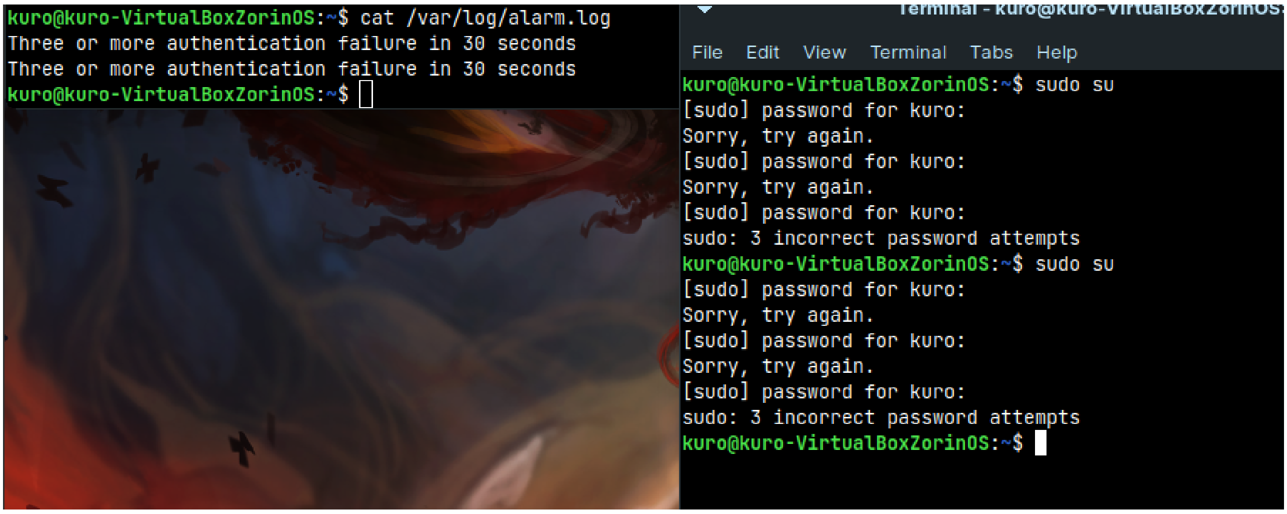


Terminal - kuro@kuro-VirtualBoxZorinOS: ~
File Edit View Terminal Tabs Help

```
#!/bin/bash

tail -n0 -f /var/log/auth.log |
grep --line-buffered "3 incorrect password attempts" |
while read ; do echo "Three or more authentication failure in 30 seconds" >>
 /var/log/alarm.log ; done
~
~
~
~
~
~
~
~
~
~
"script.sh" 5L, 204C                                          5,75          All
```

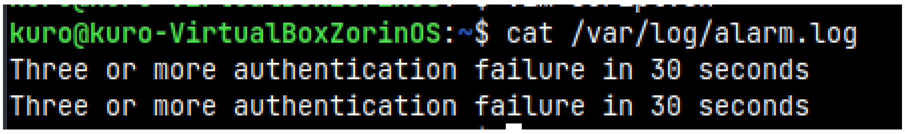
And the script in action:



Terminal - kuro@kuro-VirtualBoxZorinOS:
File Edit View Terminal Tabs Help

```
kuro@kuro-VirtualBoxZorinOS:~$ cat /var/log/alarm.log
Three or more authentication failure in 30 seconds
Three or more authentication failure in 30 seconds
kuro@kuro-VirtualBoxZorinOS:~$ 
```

```
kuro@kuro-VirtualBoxZorinOS:~$ sudo su
[sudo] password for kuro:
Sorry, try again.
[sudo] password for kuro:
Sorry, try again.
[sudo] password for kuro:
sudo: 3 incorrect password attempts
kuro@kuro-VirtualBoxZorinOS:~$ sudo su
[sudo] password for kuro:
Sorry, try again.
[sudo] password for kuro:
Sorry, try again.
[sudo] password for kuro:
sudo: 3 incorrect password attempts
kuro@kuro-VirtualBoxZorinOS:~$ 
```



```
kuro@kuro-VirtualBoxZorinOS:~$ cat /var/log/alarm.log
Three or more authentication failure in 30 seconds
Three or more authentication failure in 30 seconds
```

5. We first add the following line to our `bashrc` file, found in `~/.bashrc`:

```
export PROMPT_COMMAND='RETRN_VAL=$?;logger -p local6.debug "$(whoami) [$$]: $(history 1 | sed "s/^[ ]*[0-9]\+[ ]*//" ) [$RETRN_VAL]"'
```

Then, we log everything from `local6.*` to `/var/log/commands.log` by editing `/etc/rsyslog.d/bash.conf` and adding:

```
local6.*    /var/log/commands.log
```

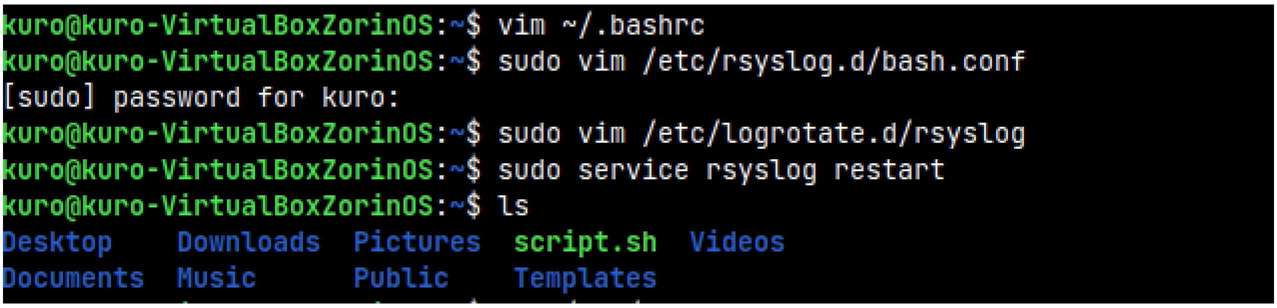
We then edit `/etc/logrotate.d/rsyslog` to rotate the logs by adding the following line:

```
/var/log/commands.log
```

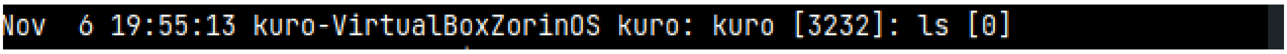
We restart `rsyslog`:

```
sudo service rsyslog restart
```

And then log out and back into the system in order for our changes to take effect:

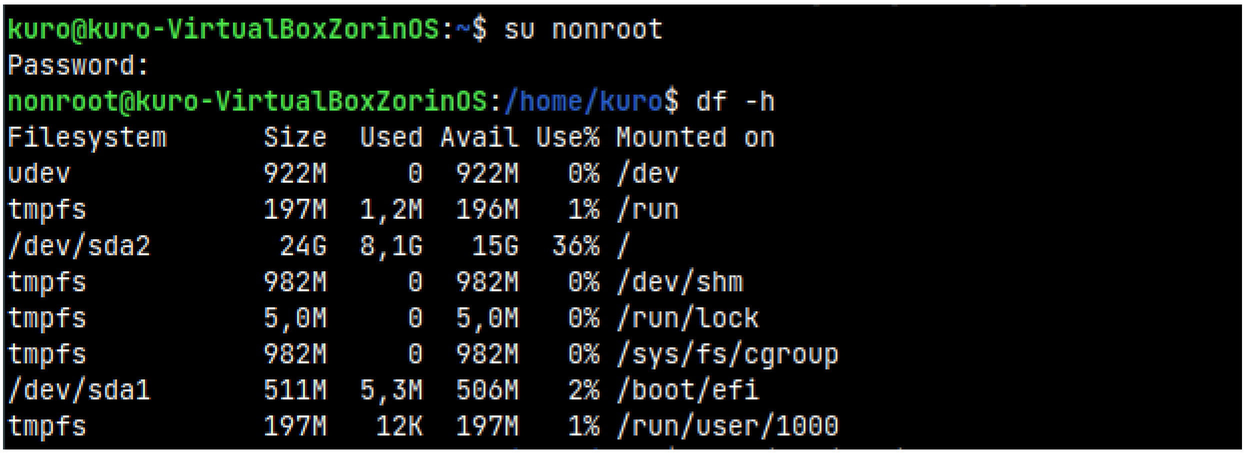


```
kuro@kuro-VirtualBoxZorinOS:~$ vim ~/.bashrc
kuro@kuro-VirtualBoxZorinOS:~$ sudo vim /etc/rsyslog.d/bash.conf
[sudo] password for kuro:
kuro@kuro-VirtualBoxZorinOS:~$ sudo vim /etc/logrotate.d/rsyslog
kuro@kuro-VirtualBoxZorinOS:~$ sudo service rsyslog restart
kuro@kuro-VirtualBoxZorinOS:~$ ls
Desktop    Downloads  Pictures  script.sh  Videos
Documents  Music      Public    Templates
```

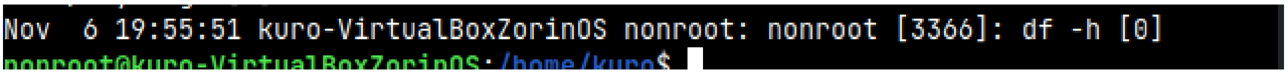


```
Nov  6 19:55:13 kuro-VirtualBoxZorinOS kuro: kuro [3232]: ls [0]
```

We can switch to another user and try a different command:

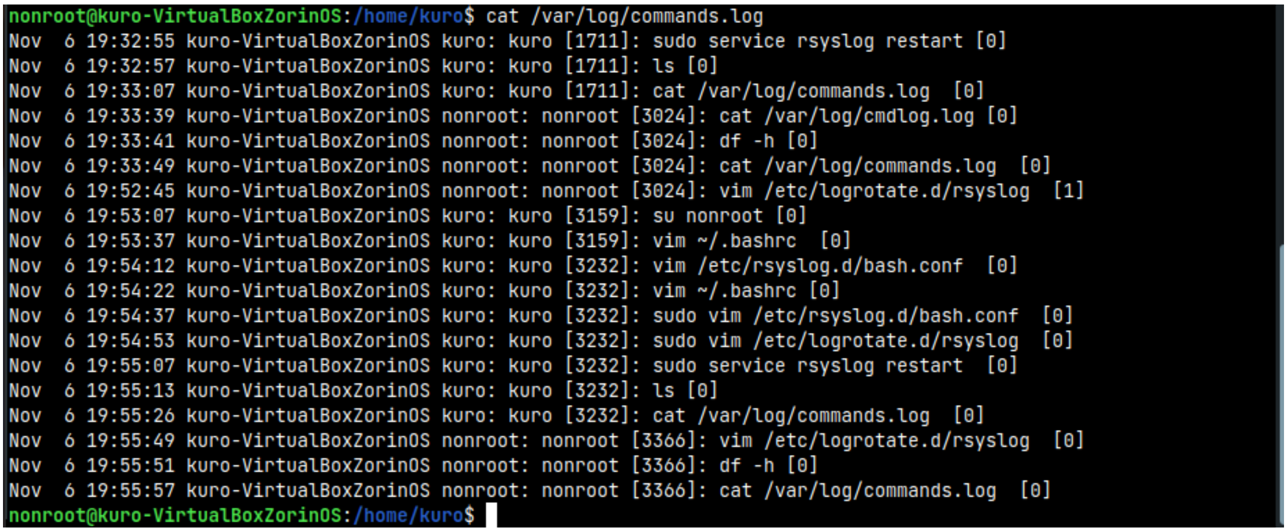


```
kuro@kuro-VirtualBoxZorinOS:~$ su nonroot
Password:
nonroot@kuro-VirtualBoxZorinOS:/home/kuro$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            922M     0  922M   0% /dev
tmpfs           197M  1,2M  196M   1% /run
/dev/sda2        24G  8,1G   15G  36% /
tmpfs           982M     0  982M   0% /dev/shm
tmpfs           5,0M     0  5,0M   0% /run/lock
tmpfs           982M     0  982M   0% /sys/fs/cgroup
/dev/sda1       511M  5,3M  506M   2% /boot/efi
tmpfs           197M   12K  197M   1% /run/user/1000
```



```
Nov  6 19:55:51 kuro-VirtualBoxZorinOS nonroot: nonroot [3366]: df -h [0]
nonroot@kuro-VirtualBoxZorinOS:/home/kuro$
```

Here's what the log file looks like:



```
nonroot@kuro-VirtualBoxZorinOS:/home/kuro$ cat /var/log/commands.log
Nov  6 19:32:55 kuro-VirtualBoxZorinOS kuro: kuro [1711]: sudo service rsyslog restart [0]
Nov  6 19:32:57 kuro-VirtualBoxZorinOS kuro: kuro [1711]: ls [0]
Nov  6 19:33:07 kuro-VirtualBoxZorinOS kuro: kuro [1711]: cat /var/log/commands.log  [0]
Nov  6 19:33:39 kuro-VirtualBoxZorinOS nonroot: nonroot [3024]: cat /var/log/cmdlog.log [0]
Nov  6 19:33:41 kuro-VirtualBoxZorinOS nonroot: nonroot [3024]: df -h [0]
Nov  6 19:33:49 kuro-VirtualBoxZorinOS nonroot: nonroot [3024]: cat /var/log/commands.log  [0]
Nov  6 19:52:45 kuro-VirtualBoxZorinOS nonroot: nonroot [3024]: vim /etc/logrotate.d/rsyslog  [1]
Nov  6 19:53:07 kuro-VirtualBoxZorinOS kuro: kuro [3159]: su nonroot [0]
Nov  6 19:53:37 kuro-VirtualBoxZorinOS kuro: kuro [3159]: vim ~/.bashrc  [0]
Nov  6 19:54:12 kuro-VirtualBoxZorinOS kuro: kuro [3232]: vim /etc/rsyslog.d/bash.conf  [0]
Nov  6 19:54:22 kuro-VirtualBoxZorinOS kuro: kuro [3232]: vim ~/.bashrc [0]
Nov  6 19:54:37 kuro-VirtualBoxZorinOS kuro: kuro [3232]: sudo vim /etc/rsyslog.d/bash.conf  [0]
Nov  6 19:54:53 kuro-VirtualBoxZorinOS kuro: kuro [3232]: sudo vim /etc/logrotate.d/rsyslog  [0]
Nov  6 19:55:07 kuro-VirtualBoxZorinOS kuro: kuro [3232]: sudo service rsyslog restart  [0]
Nov  6 19:55:13 kuro-VirtualBoxZorinOS kuro: kuro [3232]: ls [0]
Nov  6 19:55:26 kuro-VirtualBoxZorinOS kuro: kuro [3232]: cat /var/log/commands.log  [0]
Nov  6 19:55:49 kuro-VirtualBoxZorinOS nonroot: nonroot [3366]: vim /etc/logrotate.d/rsyslog  [0]
Nov  6 19:55:51 kuro-VirtualBoxZorinOS nonroot: nonroot [3366]: df -h [0]
Nov  6 19:55:57 kuro-VirtualBoxZorinOS nonroot: nonroot [3366]: cat /var/log/commands.log  [0]
nonroot@kuro-VirtualBoxZorinOS:/home/kuro$
```

The idea behind this is that `bash` provides an environment variable called `PROMPT_COMMAND`. The contents of this variable are executed as a regular `bash` command just before `bash` displays a prompt.

We can then use `logger` to log whatever info we want to see. In this case, we save the return code to a variable `RETRN_VAL=$?`,`whoami` for the current user name, the PID of the current shell, and `history` to grab the last executed command, in conjunction with `sed` which is used to remove the command index number and the whitespaces outputted by the history command, and finally the return code at the end in square brackets.

End of Exercises

Resources:

- https://askubuntu.com/questions/93566/how-to-log-all-bash-commands-by-all-users-on-a-server/93570#93570
- http://blog.kxr.me/2012/01/logging-shell-commands-in-linux.html