**1. User administration**

Question One:

*Give Jill the correct home directory*

*Jill has been given quentin’s home directory. When she logs in her bash process tries to cd to /home/quentin and fails because of permission settings. As a result the process ends up using ‘/’ or root as its current working directory.*

*Solution:*

e.g. usermod -d /home/jill jill

Question Two:

If you look at the shadow file, jill’s entry has the maximu time between password changes as less than the minimum. That cannot be right, especially when the maximum is set to 0 days.

*Change the maximum number of days between password resets so it is no longer zero*

e.g. passwd -x 30 jill

Question Three:

This is a case of having two entries in /etc/group with the same name but different group ids. That counts as two separate groups but, confusingly, with the same name.

*Change the name of group 9999 to group9999, or some other name. It now becomes clear which files jill can access.*

Question Four:

Jill cannot login because her account is locked.

*Unlock Jill’s account*

e.g. passwd -u jill

**2. Log Files Management**

Question One:

The rsyslogd process is not running. So nothing can be logged.

*Restart system logging*

e.g. systemctl start rsyslog 2>/dev/null

Question Two:

Ensure that authpriv.none is set for /var/log/messages

The problem was that authpriv.emerg failed to filter out authpriv messages, consequently they are, inappropriately allowed to reach /var/log/messages.

The correct entry is

\*.info;mail.none;**authpriv.none**;cron.none /var/log/messages

Question Three:

The problem ins that rsyslogd.conf is mis-configured. It prevents cron from logging anything.

cron.**none** /var/log/cron

*Modify /etc/rsyslog.conf to ensure that cron facility logs the appropriate events*

e.g. cron.\* /var/log/cron

You will need to restart rsyslogd

Question Four:

The error messages clearly point to a lack of a file called index.html, yet there is a file called index.htm. Index.htm is often used in a windows context.

*Rename /web/index.htm to /web/index.html (i.e. ending in ‘l’) It now works.*

Question Five:

The httpd configuration file has gone missing altogether. This is made clear in the error messages.

An investigation reveals a file called was-httpd.conf. This is perhaps an older version, but using it will get the system up and running.

*Rename /etc/httpd/conf/****was-httpd.conf*** *to /etc/httpd/conf/****httpd.conf***

**3. Process Management**

Question One:

The CPU greedy process is easily identified with top.

*Kill the appropriate process, as identified by top. No need to use ‘-9’*

Question Two:

The CPU greedy process is easily identified with top.

*Kill the appropriate process, as identified by top This time,* ***use*** *‘-9’*

Question Three:

The problem is that the system has beet set up to reboot every 5 minutes using cron.

crontab -l shows root’s crontable with a job that is set to run every five minutes.

The command in question reboots the system

*Comment out the Cron job which kicks off the 5 minutely reboots*

Question Four:

Either crond is misconfigured or it is not running. A quick check confirms crond is dead.

systemctl status crond

You need to start crond again.

*systemctl start crond*

**5. Storage**

Question One:

When creating passwords people often use numbers in place of vowels. So, the letter O instead of 0 (zero). Interchanging these characters can become a habit. This may have happened with whoever set up /etc/fstab. The error message denies the mountpoint exists when it clearly does. It’s a question of hav ing shartp eyes with this one.

*Correct a typo in /etc/fstab. DS12O1 (letter O) should read DS1201 (zero)*

*Run mount /DS1201*

Question Two:

# mount /DS1201

mount: /etc/fstab: parse error: ignore entry at line 17.

mount**: wrong fs type**, bad option, bad superblock on /dev/mapper/rootvg-Ora01,

missing codepage or helper program, or other error

In some cases useful info is found in syslog - try

dmesg | tail or so.

The error message says loud and clear, that it maybe the filesystem is of the ‘wrong type’. That means that there is a mismatch between the type of filesystem we are expecting, as per the /etc/fstab entry and the actual filesystem type in the block device (logical volume or disk slice).

# grep /DS1201 /etc/fstab

/dev/rootvg/Ora01 /DS1201 xfs defaults 0 0

*The /etc/fstab file is expecting an xfs filesystem*

*Use lsblk --fs to find out the filesystem type within Ora01*

# lsblk --fs

NAME FSTYPE LABEL UUID MOUNTPOINT

sda

├─sda1 ext4 a3bad21f-7a80-4571-80a4-cc0e008dea33 /boot

├─sda2 LVM2\_member N7yEUy-RWiJ-xbte-v8db-vFff-E6LY-yi3ruL

│ ├─rootvg-tmplv ext4 afca841b-d204-40d5-8e11-8a33dedef055 /tmp

│ ├─rootvg-usrlv ext4 7e8e2436-9391-40aa-a9db-6127f059d82e /usr

│ ├─rootvg-optlv ext4 a37e87bf-b2fd-42d2-adb5-70f36adea0a2 /opt

│ ├─rootvg-homelv ext4 52b256be-79bd-426f-a038-2bd3cb92bd4c /home

│ ├─rootvg-varlv ext4 1381a968-243d-40ec-86c4-ebf70a5e4165 /var

│ ├─rootvg-rootlv ext4 71551258-8a9e-46df-95b5-08ca80829986 /

│ └─**rootvg-Ora01 ext4** f803e793-266b-432e-ae9b-593b1d85cf6e

├─sda14

└─sda15 vfat 861B-4150 /boot/efi

sdb

└─sdb1 ext4 b806ce55-dc9e-47a4-b3a3-28162b46828f /mnt/resource

Change /etc/fstab to reflect that the Ora01 logical volume contains an ext4 filesystem and all will be well.

Question Three:

]# mount /data

mount: /etc/fstab: parse error: ignore entry at line 17.

mount: wrong fs type, bad option, bad superblock on /dev/mapper/rootvg-smallVol,

missing codepage or helper program, or other error

In some cases useful info is found in syslog - try

dmesg | tail or so.

Could be the wrong fs type again.

# grep /data /etc/fstab

/dev/rootvg/smallVol /data xfs defaults 0 0

# lsblk --fs /dev/rootvg/smallVol

NAME FSTYPE LABEL UUID MOUNTPOINT

rootvg-smallVol

There is no filesystem in the volume! The system was expecting an xfs filesystem. However we know there was once a filesystem there. maybe it is damaged.

**# xfs\_repair /dev/rootvg/smallVol**

Phase 1 - find and verify superblock...

bad primary superblock - bad magic number !!!

attempting to find secondary superblock...

..........found candidate secondary superblock...

verified secondary superblock...

writing modified primary superblock

- reporting progress in intervals of 15 minutes

Phase 2 - using internal log

- zero log...

- scan filesystem freespace and inode maps...

Metadata CRC error detected at xfs\_agf block 0x1/0x200

Metadata CRC error detected at xfs\_agi block 0x2/0x200

bad magic # 0x0 for agf 0

bad version # 0 for agf 0

bad length 0 for agf 0, should be 6400

bad uuid 00000000-0000-0000-0000-000000000000 for agf 0

bad magic # 0x0 for agi 0

bad version # 0 for agi 0

bad length # 0 for agi 0, should be 6400

*# mount /data - it works*

Question Four

**6. Advanced permissions**

Question 1

The specialCat executable will only work of the SUID bit is set. Q1 simply removes SUID from specialCat.

The solution is to put it back

chmod 4755 /scenariolabs/AdvancedPermissions/specialCat

Question 2

TH /HRDocs directory needs the SGID bit set in order to work as described, that is to associate files / directories created within /home/HRDocs with the HumanRes group. The solution is simply to put that SGID back

chmod 2770 /home/HRDocs

Question 3

/tmp needs to have the sticky bit set in order to function properly and NOT let different users delete each other’s files. The solution is to put the sticky bit back.

chmod 1777 /tmp

Question 4

The /usr/bin/passwd command allows users to change their password. This entails the updating of /etc/shadow. Naturally only super user user can so this, so the passwd command has the SUID bit set. The problem is that the passwd command is missing this and so has become non functional. Solution: put it back

chmod 4755 /usr/bin/passwd

**7. SELinux**

**<…under review…>**

Question One:

**The file ownership and permissions have been altered.**

Restore DAC permissions and ownership

chown –R apache:apache /web

find /web –type f –exec chmod 644 {} \; # Changes to appropriate permission on the files. Note 640 would do.

find /web –type d –exec chmod 755 {} \; # Changes directory permissions to a usable set. Maybe 750 would do.

chmod 755 /web/cgi-bin/myscript.sh

Question Two:

The files have been given the wrong context labels. However, the file context database remains intact. All that is needed is to restore the files contexts from those in the database.

ls –Z /web/cgi-bin

semanage fcontext –list | grep ‘^/web/cgi-bin’ # to confirm contexts ‘database’ is OK

restorecon -v “/web/cgi-bin/\*”

ls –Z /web/cgi-bin

Question Three:

The context database has been ‘corrupted’ and contains incorrect entries for /web/cgi-bin. Correct the database and update.

semanage fcontext –a –t httpd\_sys\_script\_exec\_t ‘web/cgi-bin(/.\*)?’

restorecon /web/cgi-bin/\*

Question Four:

The suggested quick fix restorecon works.

The permanent solution is to change the tar command.

tar –cf /var/tmp/web.tar  becomes tar --selinux –cf /var/tmp/web.tar

tar –xf /var/tmp/web.tar becomes tar --selinux –xf /var/tmp/web.tar

Question Five:

Reload index.html holding the shift key down whilst clicking the reload icon in the browser. This should show you an updated index.html. In the page the link ‘How It Works’ should appear. Click the link… sadly it fails. This is because the new files do not have the right file contexts.

The solution is to

1. Update the contexts database:

semanage fcontexts –a –t ‘http\_sys\_content\_t ‘/web/how(/.\*)?’

1. Use this to correct the file contexts in the new directory:

restorecon /web/how/\*

This will leave the cron job (Q3b.sh) working, as restorecon will work. A chcon command on the files in the ‘how’ directory would get the website running but fail this.

If you updated the backup/restore script correctly, the contexts will be saved and restored.