Linux Zoo answers

By James G

Contents

---

intro1

Question 2: cal

Use the cal command to find out which day the 31th of December 2002 was on. cal takes two parameters, the number of the month (e.g. 1 for January, 8 for August) and the year as a 4 digit number (e.g. 1997).

Enter the name of the day in the box below. For Monday enter "Mo", Tuesday is "Tu", Wednesday is "We", etc.

Enter the day: **Tu**

Command:

```
cal 12 2002
```

Question 3: cal year

cal can also work with 1 parameter, which is the year. When given a year, it displays a calander for the whole year. Use this feature but redirect the output of cal so that the calander for 2005 is saved to a file called "yearfile" in the home directory of demo.

Command:

```
cal 2005 > yearfile
```

Question 4: ls

Use the ls command to see the hidden and normal files you have in the home directory for "demo". Count all the files and directories shown and enter the number below. Include in the counting "." and "..".

Number of files: **9**

Command:

```
ls -lah
```

Question 5: filesize

Which file in /home/demo (including hidden files) has the smallest size? Enter the size below:

Smallest size: **18**

Command:

```
ls -lah
```

Question 6: append

Use redirection to create a file called "thismonth", containing only this month's current calendar. Then use the date command to append the current date to the end of the same file.

Command:

```
cal 11 2019 > thismonth
date >> thismonth
```

Question 7: copying

Copy the file yearfile to yearfile2. Copy the file yearfile to yearfile3.

Command:

```
cp yearfile yearfile2
cp yearfile yearfile3
```

Question 8: moving

Change the name of file yearfile3 to thisyear

Command:

```
rename yearfile3 thisyear yearfile3
```

Question 9: deleting

Delete the file yearfile

Command:

```
rm yearfile
```

Question 10: big concat

Concatenate the files "thismonth", "yearfile2" and "thisyear" together to form a single big file called "bigfile". Check that your command worked using more.

Command:

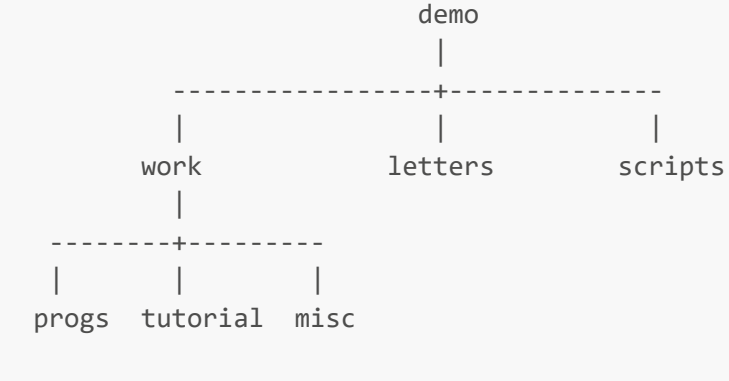
```
cat thismonth >> bigfile
cat yearfile2 >> bigfile
cat thisyear  >> bigfile
```

---

intro2

Question 2 - Create a directory structure

In the demo account /home/demo create the directory structure shown:



Commmand:

```
mkdir work
mkdir letters
mkdir scripts
cd work/
mkdir progs
mkdir tutorial
mkdir misc
```

Question 3: cp

Copy the files /etc/group and /etc/vimrc into your misc directory.

Command:

```
cp /etc/group /home/demo/work/misc
cp /etc/vimrc /home/demo/work/misc
```

Question 4: Relative mode

Make misc your current directory. Move the file vimrc to the progs directory. Use ls to verify that this worked.

Command:

```
cd work/misc
mv ./vimrc ../progs
ls ../progs
```

Question 5: rename

While still in the misc directory using a relative path name, copy the file 'bigfile' from your home directory to the tutorial directory and rename it 'bigfile2'.

Command:

```
cp /home/demo/bigfile ../tutorial/bigfile2
```

Question 6: cp

Make the directory work your current directory and using an absolute path name, copy the file 'bigfile2' in the tutorial directory to the scripts directory.

Command:

```
cd /home/demo/work
cp /home/demo/work/tutorial/bigfile2 /home/demo/scripts/
```

Question 7: tilde

We may use ~root for the home directory of the user root. Use cd and pwd to find the home directory of the user mysql.

Home directory of mysql: **/var/lib/mysql**

Command:

```
~mysql
```

Question 8: case and space

We can have spaces in our file and directory names (like Windows). File names and directory names are case sensitive (unlike Windows). Create directories with the following names in your home directories, use ls to check they are there.

My Documents gordon Gordon

```
cd home
mkdir "My Documents"
mkdir gordon
mkdir Gordon
```

Wildcard

Question 2: Wild copy

In the demo account, cd to the home directory (/home/demo). From there copy all files using the * wildcard that contain the word file in the filename into the work directory. Confirm this using ls.

Command:

```
cp *file* /home/demo/work
```

Question 3: Duplicate thismonth

Copy the file thismonth to the letters directory and rename it let1.doc. Change the current working directory to the letters directory and copy let1.doc to let2.doc. Copy once more let1.doc to let3.doc. Confirm this yourself using the ls command.

Command:

```
cp thismonth ./letters/let1.doc
cd letters/
cp let1.doc ./let2.doc
cp let1.doc ./let3.doc
ls -a
```

Question 4: Copy and rename

Copy using the ? wildcard the three files let1.doc, let2.doc, and let3.doc from the letters directory to the work/misc directory. Once copied, rename the new files (the ones in the misc directory) into rpt1.doc, rpt2.doc, and rpt3.doc. When renaming, you have to do this explicitly and not try to use a wildcard. It would appear at first that a wildcard could make the renaming easier, but it turns out to be ridiculously hard to use a wildcard in this case.

Command:

```
cp let* ../work/misc
cd ../work/misc
mv let1.doc rpt1.doc
mv let2.doc rpt2.doc
mv let3.doc rpt3.doc
```

Question 5: Square Brackets

Using square brackets, move all the files from the misc directory that contain the numbers 2 or 3 into the scripts directory.

Command:

```
mv \*[23]\* ../../scripts/
```

Question 6: rm

Remove all files beginning with the letter "r" from the scripts directory. Use the -i option to see what you are about to delete before then agreeing to delete the files involved.

Command:

```
rm r* -i
```

Question 7: Hard link

Create a hard link in progs called biglink, which is a hard link to the bigfile file in your home directory.

In "ls -l bigfile" the hard link count should be 2. If you make mistakes and create links with the wrong name or in the wrong place you will be marked wrong until you fix the problem.

Command:

```
ln /home/demo/bigfile biglink
```

Question 8: Soft link

Create a soft link in progs called mylink, which is a relative soft link to the thismonth file in your home directory.

Command:

```
ln -s ../../thismonth mylink
```

Question 9: Soft link - absolute

Create a soft link in progs called mylink2, which is a absolute soft link to the thismonth file in your home directory.

Command:

```
ln -s /home/demo/thismonth mylink2
```

Permissions

Question 2: Octal Code

What is the octal value for the permissions on /home/demo/bigfile?

Enter the octal value: **644**

Command:

```
ls -lah
```

Question 3: Other Permissions

What is the "other" permissions on /home/demo/bigfile?

Only Read

Command:

```
ls -l
```

Question 4: Remove read and execute

Take away from the scripts directory read and execute (search) permission for yourself ONLY.

Command:

```
ls -l # Obtain current permissions for group and other
```

```
chmod 255 scripts
```

Question 5: Impact of no rx

What did removing read and execute do to you ability to look into scripts?

Could not ls the directory

Command:

```
cd ./scripts # fails
ls ./scripts # fails
```

Question 6: Add read

Add in read access for yourself on the scripts directory

Command:

```
chmod +r scripts
```

Question 7: Impact of no x

What did restoring read access do to you ability to look into scripts?

Could ls, but not access contents

Command:

```
ls scripts # works
cd scripts # fails
```

Question 8: Add execute

Restore the directory scripts to read,write,execute for the owner, and read and execute for both others and group.

Command:

```
chmod +rwx scripts
```

Question 9: umask

What umask setting would be required to give, when creating a directory, read/write/execute for yourself, read and execute for group, but only execute for others?

Octal value: **026**

Pipe

Question 2: sort

Using cat create a pipe that will concatenate the two files club_members and names, sort them and send the output to the file s1.

Command:

```
cat club_members names | sort >> s1
```

Question 3: reverse sort

Using cat create a pipe that will concatenate the two files club_members and names, sort them in reverse order and send the output to the file s2.

Command:

```
cat club_members names | sort -r  >> s2
```

Question 4: sort on a field

Sort the file club_members numerically by meetings attended (this is the last field). Send the output to file s3

Command:

```
sort -nk 5 club_members >> s3
```

Question 5: sort with field seperator

Sort the /etc/passwd numerically by user id numbers (third field) in ascending order. You may wish to use the -t option here. Send the output to file s4

```
cat /etc/passwd | sort -nk 3 -t':' > s4
```

Question 6: using grep

Use grep on /usr/share/dict/words to find the first word that contains the three letter sequence wta

wta word: **cowtail**

Command:

```
grep wta /usr/share/dict/words
```

Question 7: count with grep

Use grep piped through wc on file /usr/share/dict/words to find the number of words that contain the letter x.

number of words: 12249

Command:

```
grep x /usr/share/dict/words | wc
```

Question 8: negative grep

Use grep to find all lines in /etc/passwd that do not have nologin one the line. Make grep include the line numbers of the matching lines and send the output to file s5

Command:

```
grep -nv nologin /etc/passwd > s5
```

Question 9: grep and ls

Use ls -l and grep to find all the files in the directory /etc that were last modified in August (hint: try looking for the case sensitive string "Aug"). Send this list to s6.

Command:

```
ls -lah | grep Aug > /home/demo/s6
```

Question 10: ls grep and sort

Use ls -l and grep and sort to find all the files in /etc that were last modified in Jun. Sort this list in descending order of size and then alphabetically by name (so 2 files with the same size will appear in alphabetic order). Send the output to s7. Sorting using other techniques will probably not get the same answer...

Command:

```
ls -l | grep -E Jun | sort -nrk5,5 -k9,9b > /home/demo/s7
```

Question 11: find root files

Find all the files and directories in /var (including subdirectories) that are owned by user root. Send the list of full path names to s8.

Your find command may produce "Permission Denied" errors during this search. This will have no effect on the answer, and this error can be safely ignored for this question.

Command:

```
find /var -user root > s8
```

Question 12: find .conf files

Find all the files in /etc (including subdirectories) end .conf Send the list of full path names to s9.

Your find command may produce "Permission Denied" errors during this search. This will have no effect on the answer, and this error can be safely ignored for this question.

Command:

```
find /etc -name "*.conf" > s9
```

Question 13: find new files

Find all the files and directories in the demo directory that are newer than s1. Send the output of the command to /var/tmp/t1 (don't send it to the demo directory). The names of the files should appear as full names. For example, the file "s5" would appear as "/home/demo/s5". The "/home/demo", if it appears in the output, should not have a trailing "/". The secret to avoiding the trailing slash is to use "/home/demo" not "/home/demo/" in the find command.

Command:

```
find /home/demo -newer s1 > /var/tmp/t1
```

Question 14: list large files

Find all the files in the directory directory /etc/ and its subdirectories that are larger than 1 megabyte (which you can assume is 2048 blocks of 512 bytes). Send the output to s10.

Your find command may produce "Permission Denied" errors during this search. This will have no effect on the answer, and this error can be safely ignored for this question.

Command:

```
find /etc/ -type f -size +1M > s10
```

Question 15: small xsl files

Create a directory called smallc in /home/demo. Copy into it all the file that begin with s from /usr/include that are smaller or equal to 12K. You may find these instructions on the use of find from Developer's Daily useful.

Your find command may produce "Permission Denied" errors during this search. This will have no effect on the answer, and this error can be safely ignored for this question.

Command:

```
find /usr/include -type f -not -size +12k -name 's*' -exec cp {}
```

```
/home/demo/smallc/ \;
```

---

vi

Question 2: Enter Text

In the demo account home page, use vi to edit a file called ex1. Insert into this file the following lines.

This is an exercise! Up, down, left, right, build your terminal's muscles bit by bit Once edited, you must do a write (:w then return) to write the file so that the checks can find your edits. This is true for ALL the questions in this tutorial.

Command:

```
vi ex1
[write text]
:w
:q
```

Question 4: Appending

Add " and byte by byte" to the end of the line "muscles bit by bit".

Command:

```
vi ex1
i
[write text]
:w
:q
```

Question 5: New file

Create a new file called "ex2". In this file type a number on each line from 1 to 50. Scroll up and down this file. Write the file out to disk.

Command:

```
vi ext2
[write text]
```

```
:w
:q
```

Question 6: ex2 append

Go to the end of the file and Append the following line of text:

123456789 123456789

Command:

```
vi ext2
i
[insert text]
:w
:q
```

Question 7: Cursor moves

If you are in command mode, what character are you on if you move onto the line of ex2 with "123456789 123456789" and then type

^7l Note this is HAT (e.g. shift 6) then 7 then the CHARACTER l (not a one).

8

Command:

```
vi ex2
^7l
```

Question 8: Cursor moves 2

If you are in command mode, what character are you on if you move onto the line of ex2 with "123456789 123456789" and then type

$7h

2

Command:

```
vi ex2
$7h
```

Question 9: File Search

Make sure you are in command mode, and move to the first line of ex2. Search forward for the character 8, then search on again, and then once again. What number is written on this line?

28

Command:

```
vi ex2
esc
/8
/8
/8
```

Question 10: Create ex3

Create a file using vi called ex3. Insert into this file the following lines.

Append text Insert text a computer's job is boring. Once edited, you must do a write (:w then return) to write the file so that the checks can find your edits.

Command:

```
vi ex3
[insert text]
esc
:w
:q
```

Question 11: ex3 pre insert

Add the following line of text ABOVE the first line.

First text

Command:

```
vi ex3
i
[insert text]
esc
:w
:q
```

Question 12: ex3 post append

Add the following line of text AFTER the last line

Last text

Command:

```
vi ex3
i
[insert text]
esc
:w
:q
```

Question 13: ex3 move

Move the line "Last text" to line 2 (i.e. between "First text" and "Append text")

Command:

```
vi ex3
i
[edit text]
esc
:w
:q
```

Question 14: ex3 edit

Change the last line to read "job was boring" rather than "job is boring".

Command:

```
vi ex3
i
[edit text]
esc
:w
:q
```

---

essential

Question 1: ls

Enter the number of entries in the top level directory. You can use the command ls / and simply count the words.

If you are too lazy to count you can pass the output through the wc command: ls /|wc

Enter a number: **20**

Command:

```
ls / | wc
```

Question 2: Edit

Edit the file /etc/motd so that it contains the single word Welcome You can use the echo command and redirect the output, or you can use an editor such as vi or nano (similar to pico). If you get a permission denied error ARE YOU SURE YOU ARE ROOT?

On some Windows machines, nano does not seem to like the cursor keys. If (and only if) the cursor keys result in funny characters when using nano, type the following at the prompt and then try nano again:

export TERM=vt102 This should get you going again. Silly Windows...

Command:

```
echo Welcome >> /etc/motd
```

Question 3: uid

Give the uid of the user called "operator".

Enter a number: **11**

Command:

```
id -u operator
```

Question 4: owner

Who is the owner of the directory /var/cache/httpd? If you get no information, are you sure you are using "ls" to give the directory information or the contents of the directory instead?

Enter the user name: **apache**

Command:

```
ls -la /var/cache/httpd
```

Question 5: permissions

Give the name of the first directory (alphabetically) of / that has no read permission for other.

Enter the directory name:

Command: **root**

```
ls -la /
```

Question 6: more permissions

Change the permission of the directory /var/log/httpd so that group and world have execute and read permission.

Command:

```
chmod g+xr /var/log/httpd
chmod o+xr /var/log/httpd
```

Question 7: Change owner

Change the owner of the file /etc/ntp.conf to operator

Command:

```
chown operator /etc/ntp.conf
```

Question 8: Drop zone

Create a directory /root/dropzone It should be set up so that group and other users can save files in the directory, but they cannot read the files that are there. User root must able to read and write the directory.

Command:

```
mkdir -m 733 dropzone
```

Question 9: Linking files

Create a symbolic link so that the file /usr/share/dict/words appears as /root/words

Command:

```
ln -s /usr/share/dict/words /root/words
```

Question 10: Word puzzle 1

This is a challenge question. Miss it out if you dont know regular expressions.

Use grep on words to find a word that contains each of the vowels in the correct order. How many such words are there? (you may include words with extra vowels such as adventitious. /root/words

Command:

```
grep '^.*a\+.*e\+.*i\+.*o\+.*u\+.*$' /root/words | wc
```

Question 11: Word puzzle 2

This is a challenge question. Miss it out if you dont know regular expressions.

The word minglingly includes the same four characters (e.g. ingl) repeated. How many such words are there which also begin with lower case "m" (any four character are repeated).

27

Command:

```
grep -E '^m.*(\w{4,}).*\1+.*' /root/words | wc
```

Question 12: How much space...?

Look in /usr/share/doc and find a directory starting "git-" followed by a version number. In the following questions this directory is referred to as the "git" directory...

How much space is being used by the git directory? Use a command to calculate this, and dont try adding it up yourself! We want the total answer in human readable format (eg 6.2M).

Command:

```
du -h ls /usr/share/doc | grep -E 'git-.*'
```

Question 13: How much space is available?

In human readable form (eg 123M) how much disk space is available on the main filesystem?

Command:

```
df -h /
```

Question 14

Remove the entire git directory (it is not used in our tutorials). Now repeat the above calculation for disk space, but this time give the answer in blocks.

Command

```
rm -rf git-1.8.3.1/
```

```
df /
```

---

admin

Question 1a: Partitions and LVM

Use sfdisk with block units and find out the partitions which exist in /dev/sda. How many blocks are in the first partition?

Enter a number:

512000

Command:

```
sfdisk --force /dev/sda
```

Question 1b

Use the pvdisplay command of LVM to discover what physical volume (i.e. which partition) is being managed by LVM. What is the partition being used (PV Name) and what is the volume group name (VG Name)?

PV Name: **/dev/sda2**

VG Name: **centos_lvm**

Command:

```
pvdisplay
```

Question 1c

Use lvdisplay to discover information about the VG Name found in the previous question. What is the first LV Path which is using the volume group discovered in the previous question?

LV Path: /dev/centos_lvm/root

Question 1d

Using the path discovered in the previous question, look at this path in the /dev directory using a long listing ls command. Assuming this is in fact a soft link, what is the ABSOLUTE device name which this link is pointing to?

Absolute device name: **dev/dm-0**

Command:

```
# in /dev/centos_lvm

ls -lah

# observe soft link, see that device name is located ../dm-0
```

Question 1e

For mounting this logical volume, the current method is NOT to use the volume name, or even the device it points to. Instead the device mapper is used, which can support different layers (such as encryption on top of something else). This can be found in /dev/mapper.

Look in /dev/mapper, and find the soft link which points to the device file identified in the previous question. What is the relative name of this link? So if the link was /dev/mapper/gordon, the answer wanted here is "gordon".

Mapper link name: **centos_lvm-root**

Command:

```
cd /dev/mapper

ls -lah
```

Question 1f

Look in the fstab mount table. Find the line which mounts this partition via the mapper device. Where is this partition mounted?

Mount directory: **/**

Command:

```
cat /etc/fstab
```

Question 1g

One can also mount things using the filesystem block id (which is the UUID shown in fstab). What block id could you use instead of the mapper mount in this case? It is in a format like ffffff-ffff-fffff-fffff-fffffffff.

block id for mapper filesystem: **b66fdf9b-16f0-4648-9663-536881db0ab1**

Command:

```
blkid
```

Question 1h

Recall that you discovered the device file in /dev which the LVM mapper entry was soft linked to. What was the major and minor number of this device?

Major number: **253**

Minor number: **0**

Command:

```
cd /dev

ls -lah
```

Question 1i

Somewhere in the /proc filesystem there is a file which tells you how much swap space has been allocated to the computer. Find that file and then find out how big in bytes the swap space is. Hint: the information that you require is located within the /proc directory within a file .

Enter a number: **2103484**

Command:

```
cd /proc
ls -a
cat swaps
```

Question 2a: Processes and Services

What is the process id of rsyslogd? Hint: remember the 'ps aux' command?

Enter a number: **990**

Command:

```
ps aux | grep rsyslogg
```

Question 2b

Kill rsyslogd using the kill command.

Command:

```
kill 990
```

Question 2c

Using systemctl, get the status of the rsyslog service. What is the full path to the systemd configuration file which controls the rsyslog serice?

Systemd file: **/usr/lib/systemd/system/rsyslog.service**

Command:

```
statusctl status rsyslog
```

Question 2d

Look at this configuration file. Find the line which configures the environmental variables of rsyslog (EnvironmentFile). Ignoring the "=" or the "=-" if it exists, what is the environment file for this service?

Config file: **/etc/sysconfig/rsyslog**

Command:

```
cat /usr/lib/systemd/system/rsyslog.service
```

Question 2e

Restart the rsyslogd services using systemctl. Confirm it is running using the status option.

Command:

```
systemctl reload-or-restart rsyslog
systemctl status rsyslog
```

Question 3a: Control a service

Start the database. This is called mariadb. This may take a few seconds.

Command:

```
systemctl start mariadb
```

Question 3b

Using systemctl, discover the process id (PID) for the main process. Note, mariadb has many processes, so make sure you select the Main PID.

Main PID: **3394**

Command:

```
systemctl status mariadb | grep 'Main PID'
```

Question 3c

What user is the owner of this process? Use the "ps" command with the appropriate flags.

Main PID owner: **mysql**

Command:

```
ps -ef | grep mariadb
```

Question 3d

If the main pid is the parent of the database, what is the PID of the first child of this parent pid? So if you have the process name for this PID, then pstree might help here.

PID of First Child: **3552**

Command:

```
pgrep -P 3394
```

Question 3e

Set the mariadb to run next time you boot your virtual machine.

Command:

```
systemctl enable mariadb
```

Question 3f

Systemctl has an option "list-unit-files", which says which things are enabled for running at boot. How many units are enabled?

Enter a number: **73**

Command:

```
systemctl list-unit-files | grep enabled | wc
```

Question 3g

How many enabled units are socket units?

Enter a number: **10**

Command:

```
systemctl list-unit-files | grep enabled | grep socket | wc
```

Question 3h

Now set the mariadb so that it DOES NOT run next time you boot your virtual machine.

Command:

```
systemctl disable mariadb
```

---

net

Question 2a: Main Network

What is the network device name which connects you to the linuxzoo network? This will have an IPv4 address which starts "10."... Do not type in /dev when you answer this (i.e. make the answer relative to /dev).

Main network device: **ens3**

Command:

```
netstat -n
```

Question 2b

What is the broadcast address for this network connection?

Enter an ip : **10.0.2.215**

Calculation:

10.0.2.208 = network Netmask = 255.255.255.248

10.0.2.208 + 0.0.0.7 = 10.0.2.215

Question 2c

What is the netmask in quad dotted format for this network connection?

Enter an ip 00.00.00.00: **255.255.255.248**

Question 2d

What is the IP for the default route (i.e. the gateway) for your virtual machine?

IP number: **10.0.2.214**

Command:

```
ip route show
```

Question 2e

Device eth2 is connected to a network for which your virtual machine is the gateway. The network ip is 192.168.1.0 and the netmask is 255.255.255.0. Your IP should be the last valid host IP number allowed for that network.

Calculate the your IP/NETWORK configuration for eth2.

IP number 00.00.00.00/32 : **192.168.1.254/24**

Question 2f

Configure eth2 as per the previous question. Confirm operations by pinging 192.168.1.23. Make sure the broadcast address is correct too.

Command:

```
ip address add 192.168.1.254/24 broadcast 192.168.1.255 dev eth2
```

Question 2g

Device eth3 is connected to a network for which your virtual machine is the gateway. The network ip is 192.168.3.48 and the netmask is 255.255.255.240. Your IP should be the last valid host IP number allowed for that network.

Calculate the your IP/NETWORK configuration for eth3.

IP number 00.00.00.00/32 : **192.168.3.62/28**

Question 2h

Configure eth3 as per the previous question. Confirm operations by pinging 192.168.3.50. Make sure the broadcast address is correct too.

Command:

```
ip address add 192.168.3.62/28 broadcast 192.168.3.63 dev eth3
```

NOTE: If you accidently configure a port incorrectly, you can use `ip addr flush dev ethx` to reset the interface.

Question 2i

Using an "ip link" command, discover the mac address of eth3.

Mac aa:aa:aa:aa:aa:aa : **da:4d:56:e6:83:f6**

Command:

```
ip link
```

Question 2j

What is the mac address of 192.168.3.50? Hint: "/proc"...

Mac aa:aa:aa:aa:aa:aa : **de:b8:59:48:ef:3d**

Command:

```
ping 192.168.3.50

arp -a
```

Question 2k

If you were able to log onto 192.168.3.50, how would you have configured the default route?

ip route add default via **192.168.3.62** dev eth0

Question 3a: Listening services and connections

How many programs are listening on ipv4 TCP sockets on your machine. Hint: the netstat command will help you here.

Number of TCP listeners: **8**

Command:

```

```

```
netstat -ltp | grep 'tcp '| wc     # -l = listening, -t = tcp, -p = print
```

Question 3b

What is the PID of the process which is listening on UDP port 111 (sunrpc). Hint: the netstat command will help you here.

PID of listener: **615**

Command:

```
netstat -ulp | grep sunrpc # -p = PID
```

Question 3c

When you connected to linuxzoo.net via telnet or ssh, it was forwarded from 10.200.0.1 (the main linuxzoo server) to your machine's telnet or ssh server using a proxy. Find the port number on the 10.200.0.1 end of one of these proxy connections and enter it below. Hint: again try the netstat command very near the beginning of the output... It sometimes helps to use "-n", as this does not do DNS lookups.

10.200.0.1 proxy port: **27503**

Command:

```
netstat -ant | grep 10.200.0.1
```

Question 4: Traceroute: hop count

In order for your virtual machine to reach the internet, it's packets travels through a number of virtual networks. The final network node is 10.200.0.1.

Using traceroute, find out how many hops it takes to reach 10.200.0.1. Note you must use ICMP ECHO in traceroute, rather than the default. Find the right flag in the manual.

Num of hops: **2**

Command:

```
traceroute -I 10.200.0.1
```

Question 5: nmap: Open ports

Use nmap to analyse the ports open on 10.200.0.1. As the nmap command can take quite a while to run, restrict your scan to the open tcp ports between port numbers 50 to 80 inclusive. List the open port numbers you find with spaces between them in the box below (e.g. if ports 50 and 60 are open, the answer is "50 60"). The numbers in your list must be sorted (smallest number first).

IMPORTANT. Linuxzoo security may shut you down if you produce too many packets too quickly! Use the following options for nmap or you may be kicked off the system. Even with these options the scan may take quite a few seconds.

nmap 10.200.0.1 -p 50-80 --max-retries 3

Open ports: **2**

Command:

```
nmap -p 50-80 10.200.0.1 --max-retries 3
```

Question 6a: tcpdump and web requests

The tcpdump command allows us to capture all or some of the network traffic on a particular network device. In this question use the tcpdump command to capture the behaviour of a web page request emulated using lwp-request.

Capturing packets sucessfully can be tricky, especially since you have to use the same network to talk to your machine, and you dont want to capture that too... Therefore consider the following commands to perform the capturing. NOTE you only type in the characters IN BOLD.

```
$ tcpdump -vi eth0 port 80 > /tmp/log &
[1] 3123
$ lwp-request http://linuxzoo.net
...blah...blah
...blah...blah
$ kill -1 %1
[1]+ Done ....
$
```

Where it says "eth0" you must replace that with your main network connection device, which you discovered right at the beginning of this tutorial.

Now, look at the contents of /tmp/log and enter in the box below the common IP flag which appears in most packets (The information can be found between the [..] brackets immediately after the text "flags". Ignore the information after the text "Flags" e.g. look for the one all in lowercase). Enter the two letters seen in the box below.

What is the common IP flag seen in the log: **DF** (case sensitive)

Command:

```
tcpdump -vi ens3 port 80 > /tmp/log &
lwp-request http://linuxzoo.net
kill -1 %1
```

Question 6b

Each packet it split over multiple lines. A new packet starts on a fresh line, beginning with a timestamp. This should be folowed with IP and then the IP header information.

Assuming the packet is a TCP packet, the next link should include "Flags" (with a capital F) which are the TCP flags.

Find the packets involved with the TCP Fin flag. The first of those is the packet which first requested the stream be closed. What is the TCP sequence number of that packet?

Seq no: **8371**

Command:

```
cat /tmp/log | grep -E '\[F.\]'
```

SELinux Administration

Question 1a: Global Settings

Use the getenforce administrative command. What is the current setting?

Enforcing

Command:

```
getenforce
```

Question 1b

What is the absolute pathname to the selinux directory in /sys?

/sys/fs/selinux

Command:

```
find /sys | grep selinux
```

Question 1c

How does the information from getenforce compare to the related enforce status value stored in /sys?

1

Command:

```
cat /sys/fs/selinux/enforce
```

Question 1d

How many files and directories are actually in the top level of the SELinux directory in /sys?

23

Command:

```
ls -a /sys/fs/selinux | wc # -2 for . and ..
```

Question 2a: Basic Labels

Locate the syslog daemon (called rsyslogd). What is the full true pathname?

/usr/sbin/rsyslogd

Command:

```
find / | grep rsyslogd # very inefficient
```

Question 2b

What is the SELinux label of this executable rsyslogd file?

User	Role	Type	Sensitivity
system_u	**object_r**	**syslogd_exec_t**	**s0**

Command:

```
ls -Z /usr/sbin/rsyslogd
```

Question 2c

The daemon rsyslogd uses /etc/rsyslog.conf as its configuration file. What is the SELinux label of the rsyslogd configuration file?

User	Role	Type	Sensitivity
system_u	**object_r**	**syslogd_exec_t**	**s0**

Command:

```
ls -Z /etc/rsyslog.conf
```

Question 2d

Given that rsyslogd is running currently, what is the label of the process. Use the list of running processes to discover this.

User	Role	Type	Sensitivity
system_u	**system_r**	**syslogd_t**	**s0**

Question 2e

With the label of the running process, and the label of the configuration file, use sesearch to find the semantic rules to allow the process to read the configuration file. Make sure you look only for allow rules, and limit the

search to the specific source type and target type, and also limit your search to just file rules.

Save the output of this command to /root/selinux1.

Command:

```
sesearch -A -s syslogd_t -t syslog_conf_t -c file > /root/selinux1
```

Question 2f

There are other allow rules, not just ones which relate to files. Confirm the existance of one for accessing directories fo the syslog daemon's configuration file label. Use a class of "dir" to do this. Again there should be only 1.

Save the output of this command to /root/selinux2 and confirm the contents visually.

Command:

```
sesearch -A -s syslogd_t -t syslog_conf_t -c dir > /root/selinux2
```

Question 2g

What directories in the top level of /etc have this configuration type label?

Use an "ls -Z" command on /etc, and combine it with 2 greps so you locate the correct syslog configuration label while restricting your search to just directories. Save this output to /root/selinux3.

Command:

```
ls -Z /etc | grep ^d | grep syslog_conf_t > /root/selinux3
```

Question 2h

Now use the "find" command to find all files and directories in /etc which have this configuration type label. You need to use -context. HINT: -context is the whole label, so use filename-style wildcards so you only need to specify the type.

Save this output to /root/selinux4.

Command:

```
find /etc -context *:*:syslog_conf_t:* > /root/selinux4
```

Question 3: Port Rules

The syslog daemon you investigated is allowed to open a number of ports, both tcp and udp. Use the sesearch on the syslogd_t type, focusing on tcp sockets and the name_bind permission. Include the -C to better understand conditional rules.

You should ignore rules where the line begins with DT or DF. This indicates the conditional rule is currently disabled.

How many ENABLED name_bind permissions are allowed from syslogd_t.

3

Command:

```
sesearch -AC -s syslogd_t -c tcp_socket -p name_bind | grep '^ ' | wc
```

Question 3b

Use the types of the previous answer to loop up the ports associated with those types. Taking ONLY the tcp ports, make a list ordered in ascending port number, seperated by commas if necessary, and without any white space. So if all the port types together give you tcp ports 1,5, and 10, your answer would be "1,5,10".

514,601,6514

Command:

```
semanage port -l | grep -E "^(rsh)|(syslog).*tcp"
```

Question 3c

Take the highest numbered tcp port you discovered from the last question and look it up in /etc/services. What is the services name (the first column) for this port?

syslog-tls

Command:

```
cat /etc/services | grep 6514/tcp
```

Question 4a: Process Transitions

Locate the Network Manager daemon (called NetworkManager). What is the full true pathname?

/usr/sbin/NetworkManager

Command:

```
locate NetworkManager
```

Question 4b

Given that NetworkManager is running currently, what is the label of the process. Use the list of running processes to discover this

User	Role	Type	Sensitivity
system_u	**system_r**	**NetworkManager_t**	**s0**

Command:

```
ps -auxZ | grep NetworkManager
```

Question 4c

When NetworkManager runs, it executes files in /etc/NetworkManager/dispatcher.d whenever a network interface changes state. What is the label for the executables in the dispatcher.d directory?

User	Role	Type	Sensitivity
system_u	object_r	NetworkManager_initrc_exec_t	s0

Command:

```
ls -Z /etc/NetworkManager/dispatcher.d/
```

Question 4d

When the NetworkManager process type executes a file in the dispatcher.d directory, what process transition is followed? Find the one process transition which manages this. What process type do these files run as?

initrc_t

Command:

```
sesearch -T -s NetworkManager_t | grep NetworkManager_initrc_exec_t
```

SELinux2

Question 1: Basic Labelling

Create 2 directories in /root, "secure" and "protect". Set the SELinux type of secure to system_conf_t, and set the type of protect to etc_t.

Command:

```
mkdir secure protect
chcon -t system_conf_t secure
chcon -t etc_t protect
```

Question 1b

Create a file called "test1" in secure, and "test2" in protect. Look at the types of these files. How does the types of these new files get decided?

Types are inherited from the parent

Command:

```
touch secure/test1 protect/test2
```

Question 1c

Copy test1 to protect/test3. What happens to the test3 type in comparison to test1?

The type of test3 is taken from protect

Command:

```
cp secure/test1 protect/test3
ls -Z protect/
```

Question 1d

Rename secure/test1 to protect/test4. What happens to the test4 type in comparison to the type test1 was when it was in secure (system_conf_t)?

The type of test1 is moved to test4

Command:

```
mv secure/test1 protect/test4
ls -Z protect/
```

Question 1e

Use matchpathcon to find the type which would be set if you did a restorecon on protect/test2. Save the output of matchpathcon to /root/match1. What type would be set if you did do restorecon?

admin_home_t

Command:

```
matchpathcon protect/test2 > /root/match1
```

Question 1f

Use semanage and list all of the fcontext entries, grepping the list for those which start with /root. Grep through this with the restorecon type from the previous question. This should reduce the list to just 1 regular expression, i.e. the one which matchpathcon used to produce the answer above. What is that expression?

/root(/.*)?

Command:

```
semanage fcontext -l | grep ^/root | grep admin_home_t
```

Question 1g

Add a rule to semanage fcontext so that any files in /root/ which end with .bin will be set to type bin_t. Create a file /root/test.bin and do a restorecon on that file to confirm it takes on bin_t.

Command:

```
touch /root/test.bin
ls -Z /root/test.bin
semanage fcontext -a -t bin_t "/root/.*\.bin"
restorecon /root/test.bin
```

Question 2a: Boolean control

In this section we will practice accessing and using a selinux boolean.

There is a boolean called httpd_tmp_exec. Is the boolean on or off?

off

Command:

```
getsebool -a | grep httpd_tmp_exec
```

Question 2b

Change the boolean called httpd_tmp_exec to on.

Command:

```
setsebool httpd_tmp_exec on
```

Question 2c

Find out all allow rules which are switched on by setting this boolean to on. Save the output of sesearch to /root/boolrule. When you search, find all rules, unrestricted by source types.

Command:

```
sesearch -b httpd_tmp_exec -A > /root/boolrule
```

Question 3a: Auditing

Click on the button to cause a mislabelling error for httpd

Command:

Click button

Question 3b

Start httpd with systemctl. It should fail... this should cause an event.

Command:

```
systemctl start httpd
```

Question 3c

Save the AVC event to /root/event. MAKE SURE ONLY THE AVC EVENT IS SAVED, AND THERE IS ONLY 1 AVC LINE.

Command:

```
systemctl start httpd
tail /var/log/audit/audit.log | grep type=AVC > /root/event
```

Question 3d

Use the inode information from the event. What is the full pathname of the directory in the event?

/etc/httpd/conf.d

Command:

```
tail /var/log/audit/audit.log | grep type=AVC
find / -inum 860305
restorecon /etc/httpd/conf
```

Question 3e

Use restorecon on that single directory to fix the label. Confirm that httpd now starts.

Command:

```
restorecon /etc/httpd/conf
systemctl restart httpd
```

---

fwall

Question 1a: Firewall: Empty the Chains

In this tutorial we are going to work on the firewall configuration of your machine. Some care must be taken when doing this, or you will suddenly find you can no longer log in!

In all these cases the easiest way to do the experiment is to CREATE an executable program in /root called "firewall". You should make the contents of this something like:

```
#!/bin/bash
#
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
#
# Accept ongoing connections
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
#
# For your own safety, stop users logging in from other VMs
#
iptables -A INPUT -m conntrack --ctstate NEW -m tcp -p tcp --dport 22 ! -s
10.0.0.0/16 -j ACCEPT
iptables -A INPUT -m conntrack --ctstate NEW -m tcp -p tcp --dport 23 ! -s
10.0.0.0/16 -j ACCEPT
```

```
#
# Your changes go after here.
#
```

To execute this file, remember "chmod +x ./filewall" and then just "./firewall" or "/root/firewall" to run it. Execute it once and they press the check button to make sure everything is set up ok.

After executing this file you can use "iptables -L" to show you what rules have been stored in the kernel. The provided rule uses a default policy of ACCEPT, just in case you make a typo and lock yourself out. Default ACCEPT is a bad idea here, and this is just to get you going. Later on we look at the default policy of DROP.

NOTE: If at any time you mess up and can no longer connect via telnet or ssh, then you can either reboot or use VNC to fix the problem. The script is not reexecuted automatically when you reboot, and should still be there on the next boot. VNC is much faster. If you use VNC, once logged in, you can restore the default firewall by doing "systemctl restart iptables.service".

Commmand:

```
nano /root/firewall

# paste in above configuration

chmod +x /root/firewall
/root/firewall
iptables -L
```

NOTE: To restore the default firewall, use `systemctl restart iptables.service`

Question 1b

Make the default policy of the INPUT chain DROP. Leave the other chains as they are. Do this by editing the script appropriately, then rerunning the script. Just set the INPUT chain default once in the script.

Command:

```
nano ./firewall

iptables -P INPUT ACCEPT -> iptables -P INPUT DROP
```

Question 1c

What is the device name for your primary network connection in your virtual machine. Hint: this is the one which is mentioned in your default route.

ens3

Command:

```
ip route
```

Question 1d

Visit the firewall test page, which can be found as a link off the VM Management page, and run a test on 22,23,25, and 80. All will either be "open" (service there and no firewall) or "closed" (no service there and no firewall).

Add to the END of your /root/firewall script a rule which, when an http packet (tcp) comes in from your main interface, jumps to DROP. Execute the script to activate this change. Do not make this new rule stateful (so no conntrack).

Validate this with the firewall test (which should now say "filtered").

Command:

```
nano /root/firewall

iptables -A INPUT -p tcp -i ens3 --dport 80 -j DROP
```

Question 1e

Add another rule to the end of your /root/firewall script This new rule jumps to DROP when a tcp packet which has a source address of 20.0.0.0/24 comes in from your main network device. Execute the script to activate this change. You will be marked wrong if your rule has more conditions than those listed in the question. Do not use connection tracking.

Command:

```
nano /root/firewall

iptables -A INPUT -p tcp -s 20.0.0.0/24 -i ens3 -j DROP
```

Question 1f

If any packet is passed through the FORWARD chain, reject the packet with the default settings.

Command:

```
nano /root/firewall

iptables -A FORWARD -j REJECT
```

Question 1g

Accept PING at a limit of 1 per second from any interface in the INPUT chain. Do not use connection tracking in the rule. DROP pings faster than the limit.

Warning. Even without conntrack specifically in the rule, you are still making us of connection tracking rules elsewhere in the file. Only the first ICMP ping is NEW, and the rest are RELATED. Therefore you MUST insert your new rules BEFORE the RELATED,ESTABLISHED test, and make sure that unwanted pings never reach the RELATED test (needs 2 rules).

Double check that this is working using the ping option of the firewall tests (via the "useful" tab in the control panel). You should see "limited,1/second" if you have done this correctly.

Command:

```
nano /root/firewall

iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/second -j
ACCEPT

iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Question 1h

Continue on from the previous set of rules. Add in one more rule so that if you receive pings faster than 1 per second, those pings will be logged. Note that things getting logged will appear at the end of /var/log/messages. Do not use a new chain to do this, and keep the rule as simple as possible. Do not use connection tracking.

Command:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j LOG
```

Question 1i

10.200.0.1

Question 1 - final script

```
#!/bin/bash

#

iptables -F INPUT

iptables -F OUTPUT

iptables -F FORWARD

# Question 1b

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

iptables -P FORWARD DROP

#

# Question 1g

iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/second -j
ACCEPT

# Question 1h

iptables -A INPUT -p icmp --icmp-type echo-request -j LOG

# Question 1g

iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

# Accept ongoing connections

iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

#

# For your own safety, stop users logging in from other VMs

#
```

```
iptables -A INPUT -m conntrack --ctstate NEW -m tcp -p tcp --dport 22 ! -s
10.0.0.0/16 -j ACCEPT

iptables -A INPUT -m conntrack --ctstate NEW -m tcp -p tcp --dport 23 ! -s
10.0.0.0/16 -j ACCEPT

# Your changes go after here.

#

# Question 1d

iptables -A INPUT -p tcp -i ens3 --dport 80 -j DROP

# Question 1e

iptables -A INPUT -p tcp -s 20.0.0.0/24 -i ens3 -j DROP

# Question 1f

iptables -A FORWARD -j REJECT
```

Question 2a: Firewall: Tighter Ruleset

In this tutorial we are going to work on a strict firewall configuration of your machine. Extra-special care must be taken when doing this, or you will suddenly find you can no longer log in!

Create the following as a script called firewall2 in /root.

```
#!/bin/bash
#
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
#
#
iptables -A INPUT  -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#
# --- Put a rule here if you want to be inserting at the start of INPUT
#
# ---
#
# Make sure ssh and telnet stay working, and that users on
# other VMs cannot log in.
```

```
#
iptables -A INPUT -m conntrack --ctstate NEW -p tcp --dport ssh ! -s 10.0.0.0/16 -
j ACCEPT
iptables -A INPUT -m conntrack --ctstate NEW -p tcp --dport telnet ! -s
10.0.0.0/16 -j ACCEPT
#
```

Command:

```
nano /root/firewall2

# copy and paste above configuration

chmod +x /root/firewall2

/root/firewall2
```

Question 2b

Your ssh and telnet connections are proxied from 10.200.0.1. Thus when you telnet or ssh into linuxzoo, a special program catches this traffic and forwards it on your behalf. This makes ssh and telnet traffic appear to arrive at your virtual machine from a single networked address, the proxy address of 10.200.0.1.

Insert a single rule using this script, inserting at the start of the INPUT chain so that telnet is ONLY permitted from 10.200.0.1 arriving on and device. Leave the other rules shown above unchanged. If telnet connections arrive from anywhere else, they should be directed to REJECT. Make this a stateful rule, and check the state before any other test.

Note that if you change one of the other rules, insert more than 1 rule, or do anything other than insert a single rule at the start of the chain, you will always be marked wrong!

Hint: you can put the "!" character in front of a "-s" test and the rule checks that it is NOT that address.

Command:

```
nano /root/firewall2

iptables -A INPUT -m conntrack --ctstate NEW,ESTABLISHED -p tcp --dport telnet ! -
s 10.200.0.1 -j REJECT
```

Question 2c

Add a rule to the firewall so that if someone on your virtual machine tries to open an http connection to 10.200.0.1, the packet it ACCEPTED. Add this rule to the END of the appropriate chain. Do not change any of the existing rules. Use a stateful rule, and check for NEW state before any other tests.

Command:

```
iptables -A OUTPUT -m conntrack --ctstate NEW -p tcp --dport http -d 10.200.0.1 -j
ACCEPT
```

Question 2d

Add appropriate rules so that, if this machine was a router, it would allow RELATED and ESTABLISHED traffic to flow in both directions, as well as permitting http and ping requests to an intranet machine 192.168.1.5. For your rules you can assume the intranet is on device eth9, and you should make sure that the NEW packets are sent on eth9. Do not change any rules or policy definitions from that of the previous question.

Note that in reality eth9 does not exist. However this should have no effect on the rules. Add only 1 rules for RELATED,ESTABLISHED testing, 1 rule for http, and 1 rule for ping (a total of 3 rules), and create them in that order. Make all rules stateful.

Command:

```
nano firewall2

iptables -A OUTPUT -m conntrack --ctstate NEW -p tcp --dport http -d 10.200.0.1 -j
ACCEPT

iptables -A INPUT  -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Question 2 final script

```
#!/bin/bash

#

iptables -F INPUT

iptables -F OUTPUT

iptables -F FORWARD
```

```
iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

#

# Question 2b

iptables -A INPUT -m conntrack --ctstate NEW,ESTABLISHED -p tcp --dport telnet ! -
s 10.200.0.1 -j REJECT

# Question 2c

iptables -A OUTPUT -m conntrack --ctstate NEW -p tcp --dport http -d 10.200.0.1 -j
ACCEPT

iptables -A INPUT  -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Make sure ssh and telnet stay working, and that users on

# other VMs cannot log in.

#

iptables -A INPUT -m conntrack --ctstate NEW -p tcp --dport ssh ! -s 10.0.0.0/16 -
j ACCEPT

iptables -A INPUT -m conntrack --ctstate NEW -p tcp --dport telnet ! -s
10.0.0.0/16 -j ACCEPT

#

# Question 2d

iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate NEW -p tcp --dport 80 -d 192.168.1.5 -o
eth9 -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate NEW -p icmp --icmp-type echo-request -d
192.168.1.5 -o eth9 -j ACCEPT
```

DNS

Question 1: Basic Setup

Each DNS server needs its own particular setup when working in each particular environment. In LinuxZoo, all DNS traffic is intercepted for security reasons by the gateway server and handled via a proxy. You must update the named configuration to take this into account.

Configure the /etc/named.conf file with the new options (so put this in the "options {...} area of the file) of

```
        forwarders { 10.200.0.1; };
        forward only;
```

Command:

```
nano /etc/named.conf

# paste above congifuration
```

Question 2: See it working

Start up the NAMED service (named.service) and check that it works. If you have been playing with the firewall configuration you will need to reset the firewall settings to the defaults. do that with:

systemctl restart iptables To start NAMED run "systemctl start named.service". Note this make take quite a few seconds, especially the first time, as it rebuilds its root cache. You have to reload or restart this service when you make a configuration change remember. The easiest way to check that the service and config files work is:

dig +timeout=3 +tries=1 localhost @localhost If it responds then your server is up. It should indicate that the SERVER was 127.0.0.1 (or perhaps ::1 if localhost is considered an IPv6 address) and that localhost an A record of 127.0.0.1.

Command:

```
systemctl start named.service
```

**Note: TO check that the service and config files work, use `dig +timeout=3 +tries=1 localhost @localhost`. This should indicate that the server was 127.0.0.1 and that localhost an A record of 127.0.0.1.

Question 3: New Zone

Create a brand new forward zone for domain "sillynet.net" in the named configuration directory /var/named/.

To create a new zonefile, go to /var/named and copy named.localhost to sillynet.zone (and make sure the "named" user can read the file). In this new sillynet.zone file, remove any "A" or "AAAA" lines. It should initially look something like this:

$TTL 1D @ IN SOA @ rname.invalid. (0 ; serial 1D ; refresh 1H ; retry 1W ; expire 3H) ; minimum NS @ The SOA and NS records are fine the way they are, and do not need to be changed. Add this file, along with its zone information, to the configuration file /etc/named.conf. Do not delete or mangle other information in named.conf. Keep the "." zone! By editing the sillynet.zone file, this new zone should provide:

sillynet.net -> IP address 12.0.0.20 www.sillynet.net -> IP address 12.0.0.30 Hints. It is a master type. The zone file must be readable by the user "named". Remember to use "systemctl reload named.service" when you change the config file. If you reload and it does not work, what does "systemctl -l status named.service" tell you?

Command:

```
cd /var/named

cp named.localhost sillynet.zone

nano sillynet.zone # remove specified lines

# add the following lines

# @       IN      A       12.0.0.20
# www     IN      A       12.0.0.30

chgrp named /var/named/sillynet.zone

nano /etc/named.conf

# add the following lines

# zone "sillynet.net" IN {
#         type master;
#         file "sillynet.zone";
# };
```

Question 4a: New Zone

Now you need to build a reverse zone for sillynet.zone, mapping the 12.0.0.0/24 range to these 2 new names created in the previous question. What is the zone name in in-addr.arpa format for such a zone?

Zone name:

0.0.12.in-addr.arpa

Question 4b

Now build a reverse zone for sillynet.zone, mapping the 12.0.0.0/24 range to these 2 new forward names created in the earlier question. Put their definitions into a file called "sillynet.rev".

The easiest way to create the initial format for sillynet.rev is to copy from "named.loopback". If you do use this file as your template, then remove the PTR and A and AAAA records from sillynet.rev before you start. Leave the other stuff as is. It should look something like this:

$TTL 1D @ IN SOA sillynet.net. rname.invalid. (0 ; serial 1D ; refresh 1H ; retry 1W ; expire 3H) ; minimum NS sillynet.net. In answering this question, update this new reverse zone file to produce:

12.0.0.20 -> sillynet.net 12.0.0.30 -> www.sillynet.net A nameserver definition for the zone of sillynet.net. Hints. Remember to have a zone in named.conf. Dots are important. Can the named user read the new file? Any errors in /var/log/messages?

Command:

```
cp /var/named/sillynet.zone /var/named/sillynet.rev

nano /var/named/sillynet.rev

# alter the .rev file so it looks like below

#$TTL 1D
#@       IN SOA  @ rname.invalid. (
#                                       0       ; serial
#                                       1D      ; refresh
#                                       1H      ; retry
#                                       1W      ; expire
#                                       3H )    ; minimum
#        NS      sillynet.net.
#
#20      IN      PTR     sillynet.net.
#30      IN      PTR     www.sillynet.net.

nano /etc/named.conf

# add following config to named.conf

#zone "0.0.12.in-addr.arpa" IN {
#        type master;
#        file "sillynet.rev";
#};

systemctl restart named.service
```

Question 4c

Use dig to verify the reverse lookup is operating normally. Use "dig" with the "-x" flag, to query the reverse zone for 12.0.0.30. Make sure you send this query to your own DNS server "@localhost".

In response to this dig, what is IP number returned in the A resource record information returned within the ";; ADDITIONAL SECTION" glue records?

12.0.0.20

Command:

```
dig -x 12.0.0.30 @localhost
```

Question 5: Advanced Zone

Create a brand new forward zone for domain "advanced.com". For this create a new forward zone file "advanced.zone" (copy named.localhost as a starting point), and a new reverse zone file "advanced.rev" (copy "named.loopback" as a starting point). Add both forward and reverse zones, along with the zone information, to the configuration file /etc/named.conf. The zone will use the 172.16.1.0/24 network, so in the named.conf file the reverse zone will be "1.16.172.in-addr.arpa".

This zone should give:

advanced.com -> IP address 172.16.1.1 -> MX record mail.advanced.com, priority 10 -> MX record mail.offsite.com, priority 20 www.advanced.com -> IP address 172.16.1.10, 172.16.1.11, 172.16.1.12 using a round-robin selection. mail.advanced.com -> IP address 172.16.1.1

172.16.1.1 -> advanced.com 172.16.1.10 -> www.advanced.com 172.16.1.11 -> www.advanced.com 172.16.1.12 -> www.advanced.com A nameserver definition for the reverse zone of advanced.com.

Set the responsible person email in both SOA records to me@advanced.com.

Command:

```
nano /var/named
cp sillynet.zone advanced.zone
cp sillynet.rev advanced.rev

nano advanced.zone

# edit config file so looks like below

# $TTL 1D
# @       IN SOA  @ me (
#                                       0       ; serial
#                                      1D      ; refresh
#                                       1H      ; retry
#                                       1W      ; expire
#                                       3H )    ; minimum
#        NS      @
#
```

```
#        A       172.16.1.1
#        MX      10 mail
#        MX      20 mail.offsite.com.
#
#www     A       172.16.1.10
#www     A       172.16.1.11
#www     A       172.16.1.12
#
#mail    A       172.16.1.1

nano advanced.rev

# $TTL 1D
# @       IN SOA  @ me.advanced.com. (
#                                         0       ; serial
#                                         1D      ; refresh
#                                         1H      ; retry
#                                         1W      ; expire
#                                         3H )    ; minimum
#         NS      advanced.com.
#
# 1       PTR     advanced.com.
# 10      PTR     www.advanced.com.
# 11      PTR     www.advanced.com.
# 12      PTR     www.advanced.com.

nano /etc/named.conf

#zone "advanced.com" IN {
#        type master;
#        file "advanced.zone";
#};
#
#zone "1.16.172.in-addr.arpa" IN {
#        type master;
#        file "advanced.rev";
#};

chgrp named /var/named/advanced.*

systemctl restart named
```

Diag

Question 1a: Split into 2 user groups

Click the button below to set up the scenario

Question 1b

Two users, bob and jim, currently share the same group id. User jim is being moved to another project, and he needs to be given his own group id.

Modify the user account information so that jim keeps his current user id but moves to a new group called "hoho". You have to create that new group. Make sure his home directory and all files are changed too.

This button checks to see if you have completed the task.

Command:

```
sudo groupadd hoho
usermod -g hoho jim
```

Question 1c

Once you have completed the task, click on this button to remove the scenario

Question 2a: User bill cannot log in

This button below sets up the scenario

Question 2b

User "bill" reports that when he tries to log in, the system immediately kicks him out. He tells you that his password is "bill" (i.e. the same as his username). He accidentally left himself logged in during a lab and maybe someone did something to his account?

HINT: As this is affecting only bill, check his shell (such as .bashrc or .bash_profile) files first. You can test the fault by trying to login as bill using ssh.

This button checks to see if you have completed the task.

Command:

```
cd /
nano home/bill/.bash_profile
# remove logout line
```

Question 2c

Once you have completed the task, clickcon this button to remove the scenario

Question 3a: USer Ben cannot save his work

Press the button below to set up the scenario

Question 3b

User "ben" reports that after he logs in, he is unable to create any files or directories in his home directory. He tells you that his password is "ben" (i.e. the same as his username). You will need to be familiar with the chmod command used to change file permissions. Try looking at http://www.dsl.org/cookbook/cookbook_toc.html for more info.

This button checks to see if you have completed the task.

Command:

```
cd /home
chmod 700 ben
```

Question 3c

Once you have completed the task, click on this button to remove the scenario.

Question 4a: User amy cannot run ls

The button below sets up the scenario.

Question 4b

User "amy" reports that after she logs in, the "ls" command appears to do nothing. Her password is "amy". Restore the ls command behaviour to normal.

HINT: Aliases can be used to change command behaviours.

This button checks to see if you have completed the task.

Command:

```
nano /home/amy/.bash_profile
# remove or comment alias line, making all ls operations instead echo to file
```

Question 4c

Once you have completed the task, click on this button to remove the scenario.

Apache 1

Question 1: Run the apache server

Each time you make a configuration change to the Apache server you must restart (or at the very least reload) the http service. Remember to start apache for the first time do:

systemctl start httpd.service And to reload the configuration file do: systemctl reload httpd.service Additionally if you have been changing the standard firewall configuration, you should reset the config to normal. To do this do: systemctl restart iptables.service Now get the web server running...

Command:

```
systemctl start httpd.service
```

Question 2: Add user directories

Apache allows you do have a URL which starts with /~username. This redirects the server to look for files in /home/username/public_html/. So for instance http://machine/~dave/hello.html would look for a file in /home/dave/public_html/hello.html.

To get this feature enabled you have to hunt for the configuration statements which control UserDir. Sometimes this is in /etc/httpd/conf/httpd.conf, but in Centos7 this is in /etc/httpd/conf.d/userdir.conf. Find the line:

UserDir disable Remove this line, or better yet put a # infront of it (which comments the line out). Then look on a little further on to find:

```
#UserDir public_html
```

Delete the '#' comment character from the front of this line. As you have changed the configuration remember to reload the httpd service! Now create a user called "dave", create a public_html directory in dave's home directory, and create a file hello.html in the public_html. The contents of this file should be:

```
<html>
<body>
<h1>HOST</h1>
<p>
I am clever
</p>
</body>
</html>
```

The best way to ensure that "dave" files and directories are all owned by dave is to "su - dave", make the files and directories, then CTRL-D back to root. Otherwise do "chown -R dave.dave /home/dave". The

"/home/dave" and "/home/dave/public_html" must be executable by others, and the "/home/dave/public_html/hello.html" file must be readable by other. More generous permissions or permission changes on owner or group from the default will be marked incorrect.

You can direct your browser to see this page by using the URL

http://yourmachinename/~dave/hello.html Replace "yourmachinename" with the output of running the "hostname" command, eg [root@host-19-17 dave]# hostname host-19-17.linuxzoo.net Finally, SELinux is currently enabled in enforcing mode. This means you need even more security to overcome (configure). You need to make sure that the SELinux boolean httpd_read_user_content is enabled. By default SELinux is forbidden from reading any file in /home. Check with

getsebool httpd_read_user_content and if needed set it with setsebool -P httpd_read_user_content 1 "setsebool" may take 20 or more seconds to run. It will finish, honest!

Command:

```
nano /etc/conf.d/userdir.conf
useradd dave
cd /hoee/dave
mkdir public_html
chown -R dave.dave /home/dave
cd public_html
vim hello.html
chmod/chgrp/chown # on files and folders
setsebool -P httpd_read_user_content 1
```

Question 3: Add two new directories/files

Create the following directories, each of which must be executable for other:

/home/dave/public_html/web /home/dave/public_html/vm In each of these new directories create a file called "hello.html", which are copies of hello.html from /home/dave/public_html, except in "web/hello.html" replace the word HOST with WEB. In "vm/hello.html" replace the word HOST with VM. Case is important.

Command:

```
cd /home/dave/public_html/
mkdir web
mkdir vm
cp /home/dave/public_html/hello.html /home/dave/public_html/web/hello.html
cp /home/dave/public_html/hello.html /home/dave/public_html/vm/hello.html
nano /web/hello.html
nano /vm/hello.html
```

Question 4: Create 2 virtual hosts

You need to create a number of virtual hosts in your virtual machine. These should go into a new file somewhere in /etc/httpd/conf.d. For the purposes of this tutorial, create and use the file "/etc/httpd/conf.d/zvirtual.conf".

Using create two VirtualHosts in the file /etc/httpd/conf.d/zvirtual.conf. Below is an example of a VirtualHost definition which may help you remember what is needed.

```
<VirtualHost *:80>
    ServerAdmin me@grussell.org
    DocumentRoot /home/gordon/public_html/grussell.org
    ServerName sql.grussell.org
    ErrorLog logs/sql-error_log
    CustomLog logs/sql-access_log common
</VirtualHost>
```

The names of your virtual hosts have to be worked out by yourself from your current hostname. Type in the command "hostname" and you will get something like:

host-3-2.linuxzoo.net Your machine is known by this name in DNS. It is also known by two other names, where the word "host" has been replaced with "web" and "vm". In this example of host-3-2, this machine is also known as:

web-3-2.linuxzoo.net vm-3-2.linuxzoo.net IMPORTANT: Do not just copy this example, as your machine number is likely to be entirely different. Use "hostname" and work your machine names out for yourself. Note too that your hostname can change each time you reboot, so double check each time you reboot!

Once you have your web and vm machine names, create two virtual host entries, one for each of web-?-?.linuxzoo.net and vm-?-?.linuxzoo.net, so that the DocumentRoot of web is /home/dave/public_html/web and the DocumentRoot of vm is /home/dave/public_html/vm.

Each VirtualHost tagged area (you need 2) needs to be configured, with their own ServerName and DocumentRoot. like: The other fields are not important in this question.

It is easy to make a syntax error in the config file. If you have problems you can check for syntax errors using the command:

httpd -t ...and if you make changes to a configuration file remember to tell httpd.service! Once again you can verify this works manually by pointing your browser to web-?-?.linuxzoo.net or the vm equivalent (remembering to put the right things in for the "?" characters). This is important, as in an assessment you may need to verify this yourself.

Command:

```
cd /etc/httpd/conf/conf.d/
```

```
hostname # remember for later
nano zvirtual.conf
```

```
# zvirtual.conf

<VirtualHost *:80>

    ServerAdmin 40417677@live.napier.ac.uk

    DocumentRoot /home/dave/public_html/web

    ServerName web-2-225.linuxzoo.net

</VirtualHost>


<VirtualHost *:80>

    ServerAdmin 40417677@live.napier.ac.uk

    DocumentRoot /home/dave/public_html/vm

    ServerName vm-2-225.linuxzoo.net

</VirtualHost>
```

Question 5: Rewrite Rules

Add to the VirtualHost tag area for the "vm-?-?.linuxzoo.net" virtual host a ServerAlias for "host-?-?.linuxzoo.net". Remember to replace the ? characters with the details for YOUR machine. Once again, you can remind yourself of what this is by running the hostname command. After doing this the use of host-?-? should also use the virtual host information for vm-?-?.

Add to the end of the VirtualHost tag area for vm-?-? a rewrite rule, such that any use of host-?-?.linuxzoo.net gets an external redirection to rewrite it to vm-?-?.linuxzoo.net.

Command:

```
nano zvirtual.conf

# ServerAlias host-2-225.linuxzoo.net
```

```
# zvirtual.conf

<VirtualHost *:80>

    ServerAdmin 40417677@live.napier.ac.uk

    DocumentRoot /home/dave/public_html/web

    ServerName web-2-225.linuxzoo.net

</VirtualHost>


<VirtualHost *:80>

    RewriteEngine On
    ServerAdmin 40417677@live.napier.ac.uk

    DocumentRoot /home/dave/public_html/vm

    ServerName vm-2-225.linuxzoo.net

    ServerAlias host-2-225.linuxzoo.net

    RewriteCond %{HTTP_HOST} ^host-2-225\.linuxzoo\.net$
    RewriteRule ^(.*)$ http://vm-2-225.linuxzoo.net%{REQUEST_URI} [L,R]

</VirtualHost>
```

Question 6: Extended Rewrite Conditions

Modify the rewrite rules for the previous question with an additional condition, so that host-?-?.linuxzoo.net always gets rewritten using an external redirect to vm-?-?.linuxzoo.net unless the URI starts with a /~dave. Thus:

http://host-?-?.linuxzoo.net/hello.html -> rewritten to --> http://vm-?-?.linuxzoo.net/hello.html

http://host-?-?.linuxzoo.net/~dave/hello.html -> not rewritten and handled normally

Command:

```
nano zvirtual.conf

# add RewriteCond

%{REQUEST_URI} !^/~dave/
```

apache2 / Basic Authentication

Question 1: Create TOM

Build a user called "tom" to experiment with. Use

adduser tom You also need to have the apache service (httpd) running.

Make sure your httpd.conf file supports User public_html directories. Look through the /etc/httpd/conf.d/userdir.conf file for a line:

UserDir disable Comment this line out (or delete it). Then look on a little further to find: #UserDir public_html Delete (uncomment) the '#' comment character from the front of this line. As you have changed the configuration remember to reload the httpd service! Now create a user called "tom", create a public_html directory in tom's home directory, and create a file p1.html in the public_html. The contents of this file should be:

TOM

Document body goes here.

Change the appropriate permissions on the /home/tom directories and files by the minimum amount possible to give apache permission to use the file.

Finally, SELinux is enabled in enforcing mode in Fedora 15 by default. This means you need even more security to overcome (configure). You need to make sure that the SELinux boolean httpd_read_user_content is enabled. By default SELinux is forbidden from reading any file in /home. Check with

getsebool httpd_read_user_content and if needed set it with setsebool -P httpd_read_user_content 1 "setsebool" may take 20 or more seconds to run. It will finish, honest! IN ALL CASES ENSURE tom owns ALL FILES AND DIRECTORIES in /home/tom AT ALL TIMES. Give yourself a break and do "su - tom" when you want to create stuff in /home/tom, then CTRL-D back to root for the admin stuff...

Command:

```
adduser tom
systemctl start httpd
nano /etc/httpd/conf.d/userdir.conf # comment out UserDir disable, uncomment
UserDir public_html
systemctl reload httpd
cd /home/tom
mkdir public_html
cd public_html
nano p1.html
chown tom.tom public_html/
chown tom.tom p1.html
chmod 701 tom
```

```
chmod 751 public_html/
chmod 645 p1.html
```

Question 2: Add two new directories/files

Create the following directories, each of which must be executable by others:

/home/tom/public_html/richard /home/tom/public_html/harry In each of these new directories create a file similar to p1.html, but called:

/home/tom/public_html/richard/p2.html /home/tom/public_html/harry/p3.html In "richard/p2.html" replace the word TOM with RICHARD. In "harry/p3.html" replace the word TOM with HARRY. Case is important.

Command:

```
cd /home/tom/public_html
mkdir richard
mkdir harry
cp p1.html ./richard/p2.html
cp p1.html ./harry.p3.html
nano ./richard/p2.html
nano ./harry/p3.html
```

Question 3: Basic Auth file

Create a password file for basic authentication. Remember this has nothing to do with normal unix users, and even less to do with /etc/passwd!

The htpasswd command allows you to create the file, and to add users to the file. Use it to create a basic authentication password file called "/home/tom/webpasswd". Put into this file two users with the following passwords:

```
User: richard             Password: pass1
User: harry               Password: pass2
```

Make sure that the password file is world readable.

Command:

```
htpasswd -c /home/tom/webpasswd harry
hypasswd /home/tom/webpasswd richard
```

```
chown tom.tom webpasswd
```

Question 4: Secure richard/

Secure the public_html/richard directory so only a user with the basic authentication details of richard, password pass1, can access the files.

Command:

```
> nano /home/tom/public_html/richard/.htaccess

AuthType Basic
AuthName "Richard Only"
AuthUserFile /home/tom/webpasswd
Require user richard
```

Question 5: Secure harry/

Secure the public_html/harry directory so only a user with the basic authentication details of group "magic" can access the contents.

To answer this question, create a group file "/home/tom/webgroup" with the following contents:

magic: richard harry Make sure in the .htaccess file in the harry directory you use only "Require group" and not some sort of "Require user" command. And make sure all files in /home/tom are owned by tom...

Command:

```
cd /home/tom
nano webgroup # magic: richard harry
chown tom.tom webgroup
nano /home/tom/public_html/harry/.htaccess

#AuthType Basic
#AuthName "Magic Only"
#AuthUserFile /home/tom/webpasswd
#AuthGroupFile /home/tom/webgroup

#Require group magic

systemctl reload httpd
```

Question 6: Complex requires/

1. Add to your webpasswd file an extra user, "jim", with password "walton".
2. Create a directory in public_html for "jim", called "jim", and create a file "p4.html" which is a copy of "p1.html" except you need to replace the word "TOM" with "JIM".
3. Protect this directory with basic authentication (similar to harry), but this time configure it so that that access is only permitted by either user "jim" from 10.200.0.1, or user "harry" from 127.0.0.1. Do not use any implicit requires (i.e. say RequireAny or RequireAll rather than relying on the defaults).

Command:

```
htpasswd /home/tom/webpasswd jim
cd public_html/
mkdir jim
cp p1.html jim/p4.html
nano jim/p4.html
nano jim/.htaccess


#AuthType Basic
#
#AuthName "By Invitation Only"
#
#AuthUserFile /home/tom/webpasswd
#
#AuthGroupFile /home/tom/webgroup
#
#<RequireAny>
#        <RequireAll>
#                Require ip 10.200.0.1
#                Require user jim
#        </RequireAll>
#        <RequireAll>
#                Require ip 127.0.0.1
#                Require user harry
#        </RequireAll>
#</RequireAny>
```

---

log

Question 1: Download a log

Use the command wget to download one of my server's log files. You need to do:

wget http://linuxzoo.net/data/short.log -O log This downloads one of my weblogs and saves it into a file called "log".

Command:

```
wget http://linuxzoo.net/data/short.log -O log
```

Question 2: Any 404

Look for file not found errors in this weblog. This is error 404. Although not a perfect method, you can do this by searching for " 404 " in the log. The spaces are important, otherwise a search for "404" would match "404hello" etc.

Once found save all of those to a file called "notfound".

Command:

```
cat log | grep ' 404 ' > notfound
```

Question 3: The IP numbers

Process the "notfound" file and save a list of only the IP numbers of each log entry. This can be done using

cut -f1 -d" " filename This gives the first "field" of the file "filename", where fields are delimited using the space " " character. Save that info to a file called "ip".

Command:

```
cut -f1 -d" " notfound > ip
```

Question 4: Duplicates

If you "cat ip" you will see that there are many duplicate IPs shown. The "sort -u filename" command will sort uniquely that file and remove duplicates. It also sorts the entry alpha numerically. What is the last IP printed if this uniqueness processing is applied to the ip file? Last Unique IP: **10.1.9.73**

Command:

```
sort -u ip
```

Question 5: How many times

How many times does the above IP exist in the full log file "log"? Count of Last Unique IP:

Command:

```
cat log | grep 10.1.9.73 | wc
```

Question 6: Duplicates

Save a list of the unique IP numbers from log into a file called uip.

Command:

```
cat log | cut -f1 -d" " | sort -u > uip
```

Question 7: Frequency

For each line of uip, say how many times that line occurs in log. The output should just be a single number on each line, indicating the count. Save the answer in freq. Use xargs, but make sure in your grep that the IP does not match something shorter, e.g. a line with 10.0.0.1 doesnt match a line of 10.0.0.11 (ignore the issue that the dots are unescaped). Only save the biggest 10 counts, ordered in decending order

Command:

```
cat log | cut -f1 -d" " | sort | uniq -c | awk '{print $1}' | sort -rn | head -10
> freq
```

Question 8: Frequency pt2

The problem with the freq file is that you cannot tell which IP has what frequency. Fix that by redoing the above question and making sure the output is in the format of ip,count. For example, "10.0.0.1,5". Save the answer to freq2

Hint: the xargs needs to be in the format:

```
sh -c "echo -n {},;grep -c '{} ' log"
```

So this executes a new shell, and in it does an echo (print) of the parameter, then a comma, and no return character. The semicolon marks the end of the command and the start of the next. Here do the normal grep.

Command

```
cat log | cut -f1 -d' ' | sort | uniq -c |sort -k1nr| head -10|awk '{print
$2","$1}' > freq2
```

---

mail

Question 1: Setup sendmail scenario

Click the button below to set up the sendmail scenario.

Question 2a: Aliases

Add an alias so that all email going to local user jim will instead go to local user frank.

Command:

```
nano /etc/aliases
# add jim: frank
newaliases
```

Question 2b

Add an alias so that all email going to local user bob will be silently deleted.

Command:

```
nano /etc/aliases
# add bob: /dev/null
newaliases
```

Question 2c

Add an alias so that all email going to local user peter will be sent to both allan and zoe..

Command:

```
nano /etc/aliases
# add peter: allan,zoe
newaliases
```

Question 3a: virtual hosts

Arrange sendmail so that if mail arrives for jim@null.com, it gets sent to the local user frank.

Command:

```
cd /etc/mail
nano virtusertable
# jim@null.com  frank
nano sendmail.mc
# FEATURE(virtusertable)
```

Question 3b

Keeping the previous rule for frank, make sure that all email for the null.com domain which is not jim@null.com gets delivered to local user adam.

Command:

```
nano virtusertable
# @null.com adam
```

Question 3c

Continuing on from the previous rules, now create rules so that when frank SENDS emails, then appear to come from jim@null.com. Additionally, make sure that when adam sends emails, they appear to come from adam@null.com.

Command:

```
nano genericstable
# frank jim@null.com
# adam  adam@null.com
nano sendmail.mc
```

```
# FEATURE(genericstable)
```

Question 4: debug

User iain has complained that he is not receiving any email, and instead he has heard that victor and victoria are receiving his email instead. Fix this issue for iain.

Command:

```
nano /home/iain/.forward
# remove lines
```

Question 5: spf

What would be the entry needed in the DNS information to provide an spf entry so that only the ip 10.0.0.1 was allowed to send emails for the current SOA?

Make sure the rule only contains the settings as indicated above, and make sure that the rule is set with a hard fail. Use all lower case, and only put a single space character between the elements.

@ TXT **v=spf1 ip4:10.0.0.1 -all**
