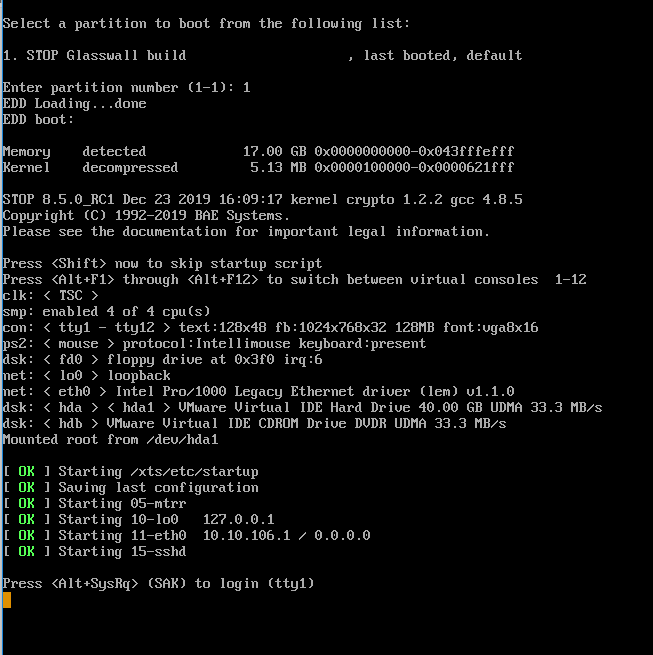
**Using STOP VM Image**

VM Image at : M:\00-Common\STOP8.5.0\_RC1\vmImage\STOP vm 2

Requires VM player 15 to run.

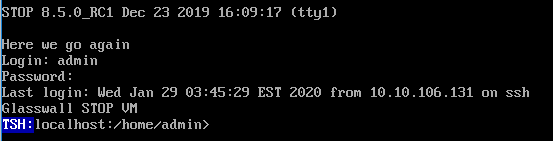
Once you start the image player it will boot up stop and you should see a login page like the below:



Hit <Alt + SysRq> to login . usedid = admin, password = Glasswall1234. (Note: On vmware you can also hit the ,Windiws. Key.)

STOP allows multiple tty sessions. Use <alt + F1> to <alt +F12> to switch between tty1- tty12.

On logging in should see something like:



You can scroll up the output in a tty session (like in command window) but first you need to hit <Scroll Lock> to enable this. Use <Scroll Lock> or <Esc> to get out of scrolling mode so you can issue commands again.

To pass control back to windows hit <Ctrl + Alt>.

If you hit <Alt + SysRq> you will be logged out.

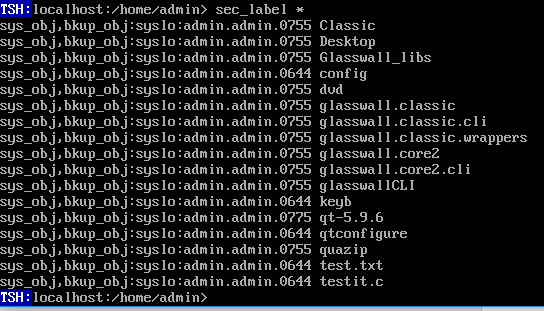
**Security labelling**

One of the fundamental differences of STOP is it’s security settings, referred to as Security labels. you are doing more than just the basics it is strongly recommended to read sections 5 and 4.2 of the manual.

Security settings are set and read using the **sec\_label** command. Sec\_label can be applied to object, e.g. files and directories and devices or processes.

**DO NOT CHANGE sec\_labels WHEN RUNNING tty SESSIONS IN XWINDOWS**

Below is an example of checking the sec-labels for the current directory:



Here is checking the sec\_label currently running processes as:



As you are mostly going to be logged in as admin then the following command will provide enough privileges to do most things:

**sec\_label –pl admin,all\_exempt**

This will need to be issued for each tty session.

**Changing labels on files**

If you need to change the labels on a file

**sec\_label –l** *filename* *label*

**Changing the label for processes**

If you need to change the labels on a file

**sec\_label –p –l** *label*

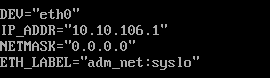
**It’s like linux but..**

It’s like linux but different enough to catch you out sometimes. Most linux commands will work as you expect.

Most OS directories that you would expect to be under ‘/’ or ‘/usr’ are in fact in ‘/xts’.

**Configuring IP address**

The IP address used by the VM is located in /xts/etc/startup.d/11-eth0 file.



Edit the file to use an IP address on the domain that is not in use. Use ping to find an IP not in use.

To make the change take effect use **shutdown –r** command to restart the OS.

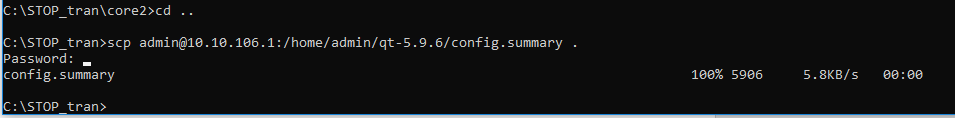
**Transferring files to/from host**

As ssh is configured using the default port scp can be used to transfer files. There is an scp client within windows 10 that can be used within the command window. You can google scp to find out all the option but the basic format needed for most occasions is:

**scp [-r]** *origin\_location desttination*

Use the –r option if copying a directory. To specify a cloation on the STOP VM it need to be in the format userid@ip\_address:/location, e.g. [admin@10.10.106.1:/home/admin/myfiles](mailto:admin@10.10.106.1:/home/admin/myfiles). You will be prompted for the password of the STOP userid.

Below is an example of copying from the STOP VM to the current directory:

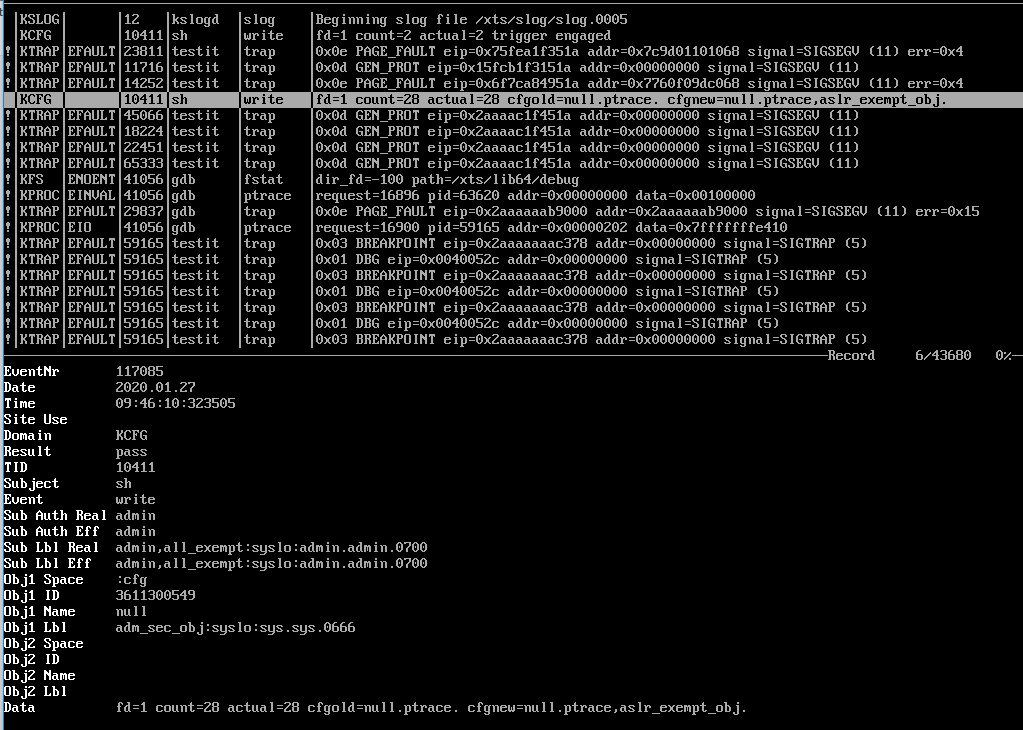


When using scp to copy large numbers of files you can find some file do not copy and you get an “eother” error on some file. This erro is known and the easiest way to get round it is to zip all the file you want to transfer and then transfer the zip file.

A more clumsy way to transfer files to STOP is to tell VM player to use an iso file as the DVD player and then mount the dvd within stop (**mount /dev/hdb /mnt/iso**).

**Viewing system logs**

Interaction with the OS are logged and these can be viewed using the **nslogfmt** tool. For every system call a record is created and this can be useful in debugging.



It can be useful to start a new log file before executing a test to make sure only tasks relevant to your test are in the log. To do this issue the following commands:

**cd /xts/cfg/slogd**

**echo 1 > path\_switch\_trigger**

**The debugger**

The gnu debugger is incorporated into STOP and can be used in debugging code. You may need to adjust your permissions to get trace information. The suggested way to do this is to temporarily add the “ptrace” permission to the null role (see below).

One other suggestion is to add the role aslr\_exempt\_obj so that programs load at the same virtual address every time

**Updating Roles**

The role definitions are located at /xts/cfg/security/role. Each role is a file of the format:

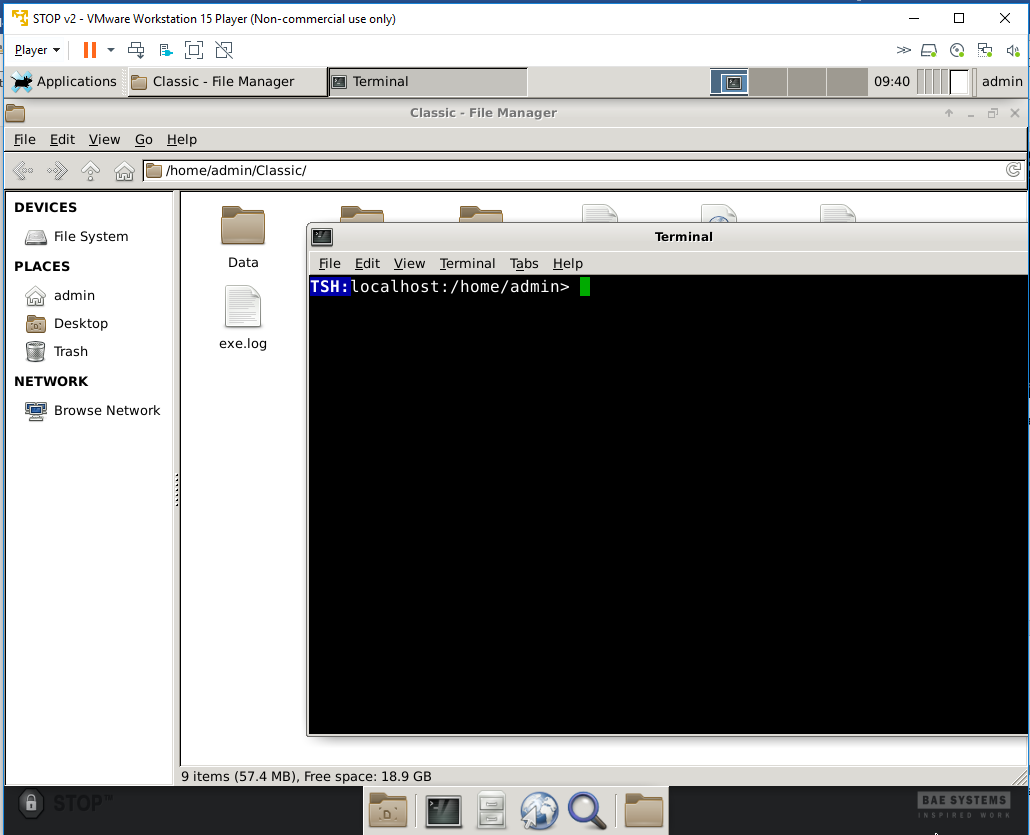
<comma separated list of roles inherits>.<comma separated additional permissions>.



Roles can be updated by editing the file with the additional roles or permissions required. To see the list of available permissions **cat /xts/cfg/security/requests**.

**Starting xwindows**

To start xwindows type **startx**



**Note:** We’ve experienced a few odd issues with mouse movements and control, especially when RDP’s to a PC running the VM.

**GLASSWALL locations**

**Classic SO**

All the files needed for Classic are located at /home/admin/glasswall.classic. Within that directory everything is located in the same place as for linux. This also includes any directories with linux in the name as we did not change them to get this working.

Note: The makefile has been altered significantly to get the compile working so beware of editing it.

**Classic CLI**

/home/admin/glasswall.classic.cli

**CLI run location + data**

/home/admin/Classic

**QT Framework**

All QT stuff at: /home/admin/qt-5.9.6

There is a copy of the compiled libraries in: /xts/qt-5.9.6

This installation uses the open source version of QT but there are linux specific calls that have been replaces with their STOP equivalent.

To recompile all the QT libraries without running a **configure**

**cd /home/admin/qt-5.9.6**

**make –j***x*(where x=number of processors)

To copy the newly compiled libraries to /xts/qt-5.9.6

**make install**

The command used to execute the configure on this installation of STOP is at **/home/admin/qt-5.9.6/README-XTS**

**Quazip**

File location: /home/admin/quazip

Steps to create new compilation of quazip

**/xts/qt-6.9.6/bin/qmake –o Makefile quazip.pro**

**make**

**Core 2**

Release build

**/xts/qt-6.9.6/bin/qmake –r –o Makefile core2.pro -spec linux-g++**

Debug build

**/xts/qt-6.9.6/bin/qmake –r –o Makefile core2.pro -spec linux-g++ CONFIG+=debug CONFIG+=qml\_debug**