How to Investigate a core dump

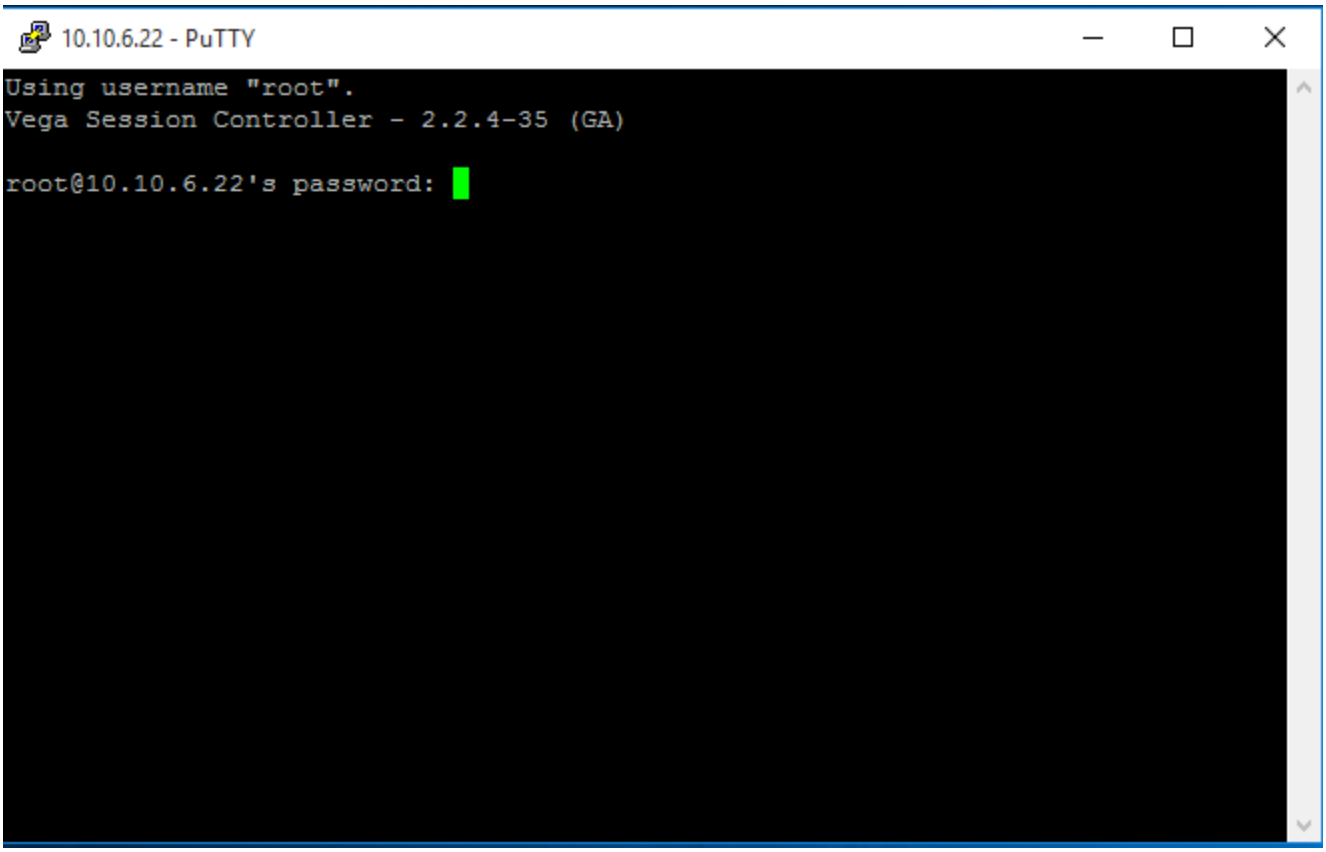
- How to check if the SBC created a 'core' file
- Opening a core file to investigatedetails

When investigating a functional problem with the Sangoma SBC (appliance and/or software version), it is a good idea to check if a 'core' file was created around the time of issue.

How to check if the SBC created a 'core' file

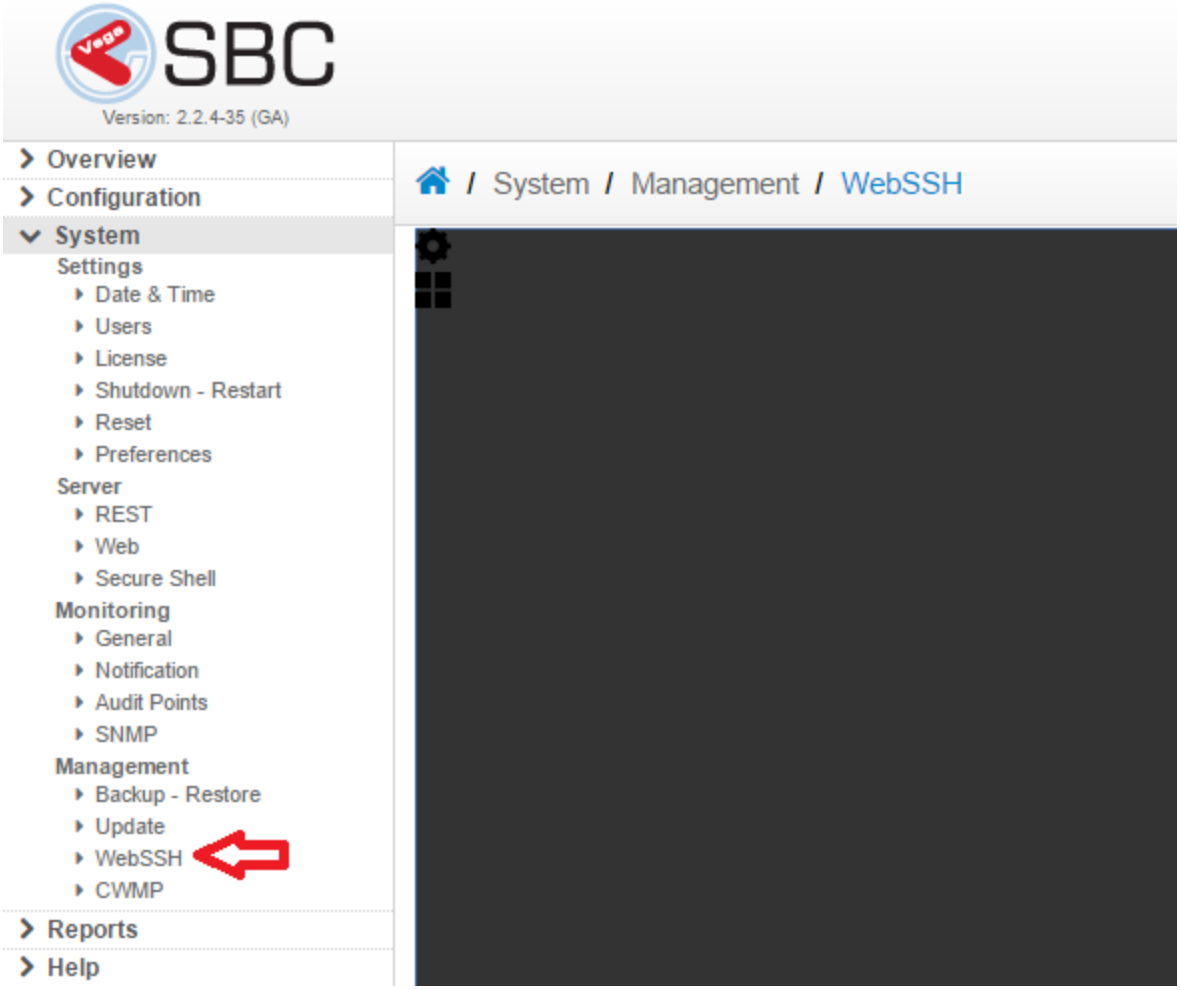
To check if your SBC has created a 'core' file, open an SSH session to the SBC.

An SSH session can be created by using either of the two methods:
1. use a terminal program (such as 'putty' on a windows box) and use your log in credentials with the IP address of the SBC



2.use the SBC'swebSSH feature which is built into thewebgui.

*note, you must use httpsto use the webssh



If a 'core' file was created, it will be located in the */var/core* directory.
To navigate to this directory simply type the following in your SSH session:
->cd /var/core

Now type *'ls -al'* to see if any 'core' files are present. This will show you any core files plus the time the core file was created.
If the time of the core file was weeks or months before the time of the issue, it is most likely not of interestto your case which means you can stopreadign this wiki page :)

The following is a screenshot of an SBC that had a core file created on December 8 2015:

```
root@csgsbc core
# ls -al
total 141464
drwxr-xr-x.  2 webconfig root      4096 Dec  8 08:58 .
drwxr-xr-x  25 root      root      4096 Nov 24 02:53 ..
-rw-r--r--   1 root      root 14485094[ Dec  8 08:58 core.nsc.11.19604.1449583109
```

Now lets go ahead and open this 'core' file for details

Opening a core file to investigatedetails

1. To investigate the previous core file created copy the complete core file name.
   i.e. "core.nsc.11.19604.1449583109"

2. Type the following string in your SSH session. (You can be in any directory when you type this)
   -> gdb /usr/local/nsc/bin/nsc <complete core file name>
   -> press <enter>

i.e

```
Loaded symbols for /usr/local/nsc/mod/mod_enum.so
Reading symbols from /usr/local/nsc/mod/mod_rad_auth.so...done.
Loaded symbols for /usr/local/nsc/mod/mod_rad_auth.so
Reading symbols from /usr/local/nsc/lib/libfreeradius-client.so.2...done.
Loaded symbols for /usr/local/nsc/lib/libfreeradius-client.so.2
Reading symbols from /usr/local/nsc/mod/mod_h26x.so...done.
Loaded symbols for /usr/local/nsc/mod/mod_h26x.so
Reading symbols from /usr/local/nsc/mod/mod_sofia.so...done.
Loaded symbols for /usr/local/nsc/mod/mod_sofia.so

warning: no loadable sections found in added symbol-file system-supplied DSO at 0x7fff7a509000
Core was generated by `/usr/local/nsc/bin/nsc -ncwait -nonat -nocal -rp'.
Program terminated with signal 11, Segmentation fault.
#0  0x00007f04cf8fbe83 in pthread_rwlock_tryrdlock () from /lib/libpthread.so.0
Missing separate debuginfos, use: debuginfo-install cyrus-sasl-lib-2.1.23-13.el6_3.1.x86_64 glibc-2.1
-1.4-4.el6.x86_64 krb5-libs-1.10.3-10.el6_4.4.x86_64 libcom_err-1.41.12-14.el6_4.2.x86_64 libcurl-7.3
2.el6.x86_64 libjpeg-turbo-1.2.1-1.el6.x86_64 libselinux-2.0.94-5.3.el6_4.1.x86_64 libssh2-1.4.2-1.el
el6.x86_64 libunwind-1.1-94.1.x86_64 libuuid-2.17.2-12.9.el6.x86_64 mysql-connector-odbc-5.1.5r1144-7
3.20090208.el6.x86_64 nsc-2.2.4-35.x86_64 nspr-4.9.2-1.el6.x86_64 nss-3.14.0.0-12.el6.x86_64 nss-softo
 openldap-2.4.23-32.v6.x86_64 openssl-1.0.1e-30.el6_6.5.x86_64 unixODBC-2.2.14-12.el6_3.x86_64 xz-libs
6_64
(gdb)
```

3. Now type 'backtrace' at the end of the output.
   i.e.

```
Missing separate debuginfos, use: debuginfo-install cyrus-sasl-lib-2.1.23-13.el6_3.1.x86_64 glibc-2.12-
64 keyutils-libs-1.4-4.el6.x86_64 krb5-libs-1.10.3-10.el6_4.4.x86_64 libcom_err-1.41.12-14.el6_4.2.x86_
-4.4.7-3.el6.x86_64 libidn-1.18-2.el6.x86_64 libjpeg-turbo-1.2.1-1.el6.x86_64 libselinux-2.0.94-5.3.el6
dc++-4.4.7-3.el6.x86_64 libtool-ltdl-2.2.6-15.5.el6.x86_64 libunwind-1.1-94.1.x86_64 libuuid-2.17.2-12.
-7.el6.x86_64 mysql-libs-5.1.69-1.el6_4.x86_64 ncurses-libs-5.7-3.20090208.el6.x86_64 nsc-2.2.4-35.x86_
16.x86_64 nss-softokn-freebl-3.12.9-11.el6.x86_64 nss-util-3.14.0.0-2.el6.x86_64 openldap-2.4.23-32.v6.
ODBC-2.2.14-12.el6_3.x86_64 xz-libs-4.999.9-0.3.beta.20091007git.el6.x86_64 zlib-1.2.3-29.el6.x86_64
(gdb) backtrace
```

4. You should see more output now. Below is an example output after typing 'backtrace'

```
(gdb) backtrace
#0  0x00007f4e1a53afd3 in tcmalloc::ThreadCache::ReleaseToCentralCache(tcmalloc::ThreadCache::FreeList*
   from /usr/lib64/libtcmalloc.so.4.2.2
#1  0x00007f4e1a53b757 in tcmalloc::ThreadCache::Scavenge() () from /usr/lib64/libtcmalloc.so.4.2.2
#2  0x00007f4e1a54a1b3 in tc_free () from /usr/lib64/libtcmalloc.so.4.2.2
#3  0x00007f4e09718ccd in _su_home_deinit (home=0x55ca780) at su_alloc.c:1123
#4  0x00007f4e09718206 in su_home_unref (home=0x55ca780) at su_alloc.c:774
#5  0x00007f4e09653673 in msg_destroy (msg=0x55ca780) at msg.c:168
#6  0x00007f4e0968129d in incoming_reclaim (irq=0x9420a00) at nta.c:5739
#7  0x00007f4e09681370 in incoming_reclaim_queued (rm=0x0, msg=0x0, u=0x7f4e0950fad0) at nta.c:5764
#8  0x00007f4e09685644 in incoming_mass_destroy (sa=0x2035b00, q=0x7f4e0950fad0) at nta.c:7130
#9  0x00007f4e0968548a in _nta_incoming_timer (sa=0x2035b00) at nta.c:7092
#10 0x00007f4e09670a71 in agent_timer (rm=0x2028b80, timer=0x1f4c260, agent=0x2035b00) at nta.c:1275
#11 0x00007f4e09714b8d in su_timer_expire (timers=0x23feb28, timeout=0x7f4e0950fc88, now=...) at su_tim
#12 0x00007f4e0971558a in su_base_port_run (self=0x23feae0) at su_base_port.c:339
#13 0x00007f4e0970f046 in su_port_run (self=0x23feae0) at su_port.h:326
#14 0x00007f4e09710133 in su_root_run (self=0x2687180) at su_root.c:819
#15 0x00007f4e09716fb1 in su_pthread_port_clone_main (varg=0x7f4e0954b370) at su_pthread_port.c:334
#16 0x00007f4e18f3e9d1 in start_thread () from /lib/libpthread.so.0
#17 0x00007f4e181cb8fd in clone () from /lib/libc.so.6
(gdb)
```

5. The output from the above 'backtrace' is what needs to be investigated. This is what Sangoma Support and Sangoma R&D want to see when investigating a 'core' file.
   Please send this output along with trouble ticket with Sangoma.
   If more output is required, you can type the following command:
   -> thread apply all bt

   This output will provide the details of all threads, not just the thread that caused the 'core' file.

Note: core file investigation is required to be performed on the SBC. If you try to copy the core fileoff the SBC andinvesgitate on another system, the investigation will not be useful.