Subject: Re: Application is being killed by kernel with signal 11
Posted by dev on Thu, 02 Nov 2006 15:54:30 GMT
View Forum Message <> Reply to Message

SIGSEGV is sent by kernel usually when application tries to access memory by invalid pointer.
The typical reasons for this are:
1. memory corruptions. check your hardware according to http://wiki.openvz.org/Hardware_testing

2. bugs in software. this can be caught by:
a) adding kernel messages when SIGSEGV is sent.
for example, for i386 arch check arch/i386/mm/fault.c, function do_page_fault():

```
<skipped>
bad_area:
	up_read(&mm->mmap_sem);

bad_area_nosemaphore:
	/* User mode accesses just cause a SIGSEGV */
	if (error_code & 4) {
		/*
		 * Valid to do another page fault here because this one came
		 * from user space.
		 */
		if (is_prefetch(regs, address, error_code))
			return;

		tsk->thread.cr2 = address;
		/* Kernel addresses are always protection faults */
		tsk->thread.error_code = error_code | (address >= TASK_SIZE);
		tsk->thread.trap_no = 14;
		info.si_signo = SIGSEGV;
		info.si_errno = 0;
		/* info.si_code has been set above */
		info.si_addr = (void __user *)address;
		force_sig_info(SIGSEGV, &info, tsk);
		return;
	}
```

SIGSEGV is sent above by force_sig_info(), so what you can do here is to print application eip address (regs->eip) to check where its problem occurs.

b) debug application with gdb to see where it gets SIGSEGV.
c) save core dump from an application and analyze it later.