Kdump for Linux Kernel Crash Analysis

Kdump is an utility used to capture the system core dump in the event of system crashes.

These captured core dumps can be used later to analyze the exact cause of the system failure and implement the necessary fix to prevent the crashes in future.

Kdump reserves a small portion of the memory for the secondary kernel called crashkernel.

This secondary or crash kernel is used the capture the core dump image whenever the system crashes.

1. Install Kdump Tools

First, install the kdump, which is part of kexec-tools package.

**# yum install kexec-tools**

2. Set crashkernel in grub.conf

Once the package is installed, edit /boot/grub/grub.conf file and set the amount of memory to be reserved for the kdump crash kernel.

You can edit the /boot/grub/grub.conf for the value crashkernel and set it to either auto or user-specified value. It is recommended to use minimum of 128M for a machine with 2G memory or higher.

In the following example, look for the line that start with “kernel”, where it is set to “crashkernel=auto”.

**# vi /boot/grub/grub.conf  
default=0  
timeout=5  
splashimage=(hd0,0)/grub/splash.xpm.gz  
hiddenmenu  
title Red Hat Enterprise Linux (2.6.32-419.el6.x86\_64)  
 root (hd0,0)  
 kernel /vmlinuz-2.6.32-419.el6.x86\_64 ro root=/dev/mapper/VolGroup-lv\_root rd\_NO\_LUKS LANG=en\_US.UTF-8 rd\_NO\_MD rd\_LVM\_LV=VolGroup/lv\_swap SYSFONT=latarcyrheb-sun16 crashkernel=auto rd\_LVM\_LV=VolGroup/lv\_root  KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_DM rhgb quiet  
 initrd /initramfs-2.6.32-419.el6.x86\_64.img**

3. Configure Dump Location

Once the kernel crashes, the core dump can be captured to local filesystem or remote filesystem(NFS) based on the settings defined in /etc/kdump.conf (in SLES operating system the path is /etc/sysconfig/kdump).

This file is automatically created when the kexec-tools package is installed.

All the entries in this file will be commented out by default. You can uncomment the ones that are needed for your best options.

**# vi /etc/kdump.conf  
#raw /dev/sda5  
#ext4 /dev/sda3  
#ext4 LABEL=/boot  
#ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937  
#net my.server.com:/export/tmp  
#net user@my.server.com  
path /var/crash  
core\_collector makedumpfile -c --message-level 1 -d 31  
#core\_collector scp  
#core\_collector cp --sparse=always  
#extra\_bins /bin/cp  
#link\_delay 60  
#kdump\_post /var/crash/scripts/kdump-post.sh  
#extra\_bins /usr/bin/lftp  
#disk\_timeout 30  
#extra\_modules gfs2  
#options modulename options  
#default shell  
#debug\_mem\_level 0  
#force\_rebuild 1  
#sshkey /root/.ssh/kdump\_id\_rsa**

In the above file:

* To write the dump to a raw device, you can uncomment “raw /dev/sda5″ and change it to point to correct dump location.
* If you want to change the path of the dump location, uncomment and change “path /var/crash” to point to the new location.
* For NFS, you can uncomment “#net my.server.com:/export/tmp” and point to the current NFS server location.

4. Configure Core Collector

The next step is to configure the core collector in Kdump configuration file. It is important to compress the data captured and filter all the unnecessary information from the captured core file.

To enable the core collector, uncomment the following line that starts with core\_collector.

**core\_collector makedumpfile -c --message-level 1 -d 31**

* makedumpfile specified in the core\_collector actually makes a small DUMPFILE by compressing the data.
* makedumpfile provides two DUMPFILE formats (the ELF format and the kdump-compressed format).
* By default, makedumpfile makes a DUMPFILE in the kdump-compressed format.
* The kdump-compressed format can be read only with the crash utility, and it can be smaller than the ELF format because of the compression support.
* The ELF format is readable with GDB and the crash utility.
* -c is to compresses dump data by each page
* -d is the number of pages that are unnecessary and can be ignored.

If you uncomment the line #default shell then the shell is invoked if the kdump fails to collect the core. Then the administrator can manually take the core dump using makedumpfile commands.

5. Restart kdump Services

Once kdump is configured, restart the kdump services,

**# service kdump restart  
Stopping kdump:   [  OK  ]  
Starting kdump:   [  OK  ]  
  
# service kdump status  
Kdump is operational**

If you have any issues in starting the services, then kdump module or crashkernel parameter has not been setup properly. So, verify /proc/cmdline and make sure it reflects to include the crashkernel value.

6. Manually Trigger the Core Dump

You can manually trigger the core dump using the following commands:

**echo 1 > /proc/sys/kernel/sysrq  
echo c > /proc/sysrq-trigger**

The server will reboot itself and the crash dump will be generated.

7. View the Core Files

Once the server is rebooted, you will see the core file is generated under /var/crash based on location defined in /var/crash.

You will see vmcore and vmcore-dmseg.txt file:

**# ls -lR /var/crash  
drwxr-xr-x. 2 root root 4096 Mar 26 11:06 127.0.0.1-2014-03-26-11:06:43  
  
/var/crash/127.0.0.1-2014-03-26-11:06:43:  
-rw-------. 1 root root 33595159 Mar 26 11:06 vmcore  
-rw-r--r--. 1 root root    79498 Mar 26 11:06 vmcore-dmesg.txt**

8. Kdump analysis using crash

Crash utility is used to analyze the core file captured by kdump.

It can also be used to analyze the core files created by other dump utilities like netdump, diskdump, xendump.

You need to ensure the “kernel-debuginfo” package is present and it is at the same level as the kernel.

Launch the crash tool as shown below. After you this command, you will get a cash prompt, where you can execute crash commands:

**# crash /var/crash/127.0.0.1-2014-03-26-12\:24\:39/vmcore /usr/lib/debug/lib/modules/ /vmlinux  
  
crash>**

9. View the Process when System Crashed

Execute ps command at the crash prompt, which will display all the [running process](http://www.thegeekstuff.com/2011/04/ps-command-examples/) when the system crashed.

**crash> ps  
  PID    PPID  CPU       TASK        ST  %MEM     VSZ    RSS  COMM  
     0      0   0  ffffffff81a8d020  RU   0.0       0      0  [swapper]  
     1      0   0  ffff88013e7db500  IN   0.0   19356   1544  init  
     2      0   0  ffff88013e7daaa0  IN   0.0       0      0  [kthreadd]  
     3      2   0  ffff88013e7da040  IN   0.0       0      0  [migration/0]  
     4      2   0  ffff88013e7e9540  IN   0.0       0      0  [ksoftirqd/0]  
     7      2   0  ffff88013dc19500  IN   0.0       0      0  [events/0]**

10. View Swap space when System Crashed

Execute swap command at the crash prompt, which will display the [swap space](http://www.thegeekstuff.com/2010/08/how-to-add-swap-space/) usage when the system crashed.

**crash> swap  
FILENAME           TYPE         SIZE      USED   PCT  PRIORITY  
/dm-1            PARTITION    2064376k       0k   0%     -1**

11. View IPCS when System Crashed

Execute ipcs command at the crash prompt, which will display the [shared memory usage](http://www.thegeekstuff.com/2010/08/ipcs-command-examples/) when the system crashed.

**crash> ipcs  
SHMID\_KERNEL     KEY      SHMID      UID   PERMS BYTES      NATTCH STATUS  
(none allocated)  
  
SEM\_ARRAY        KEY      SEMID      UID   PERMS NSEMS  
ffff8801394c0990 00000000 0          0     600   1  
ffff880138f09bd0 00000000 65537      0     600   1  
  
MSG\_QUEUE        KEY      MSQID      UID   PERMS USED-BYTES   MESSAGES  
(none allocated)**

12. View IRQ when System Crashed

Execute irq command at the crash prompt, which will display the [IRQ stats](http://www.thegeekstuff.com/2014/01/linux-interrupts/) when the system crashed.

**crash> irq -s  
          CPU0  
 0:        149  IO-APIC-edge     timer  
 1:        453  IO-APIC-edge     i8042  
 7:          0  IO-APIC-edge     parport0  
 8:          0  IO-APIC-edge     rtc0  
 9:          0  IO-APIC-fasteoi  acpi  
12:        111  IO-APIC-edge     i8042  
14:        108  IO-APIC-edge     ata\_piix  
.  
.**

vtop – This command translates a user or kernel virtual address to its physical address.

foreach – This command displays data for multiple tasks in the system

waitq – This command displays all the tasks queued on a wait queue.

13. View the Virtual Memory when System Crashed

Execute vm command at the crash prompt, which will display the virtual memory usage when the system crashed.

crash> vm  
PID: 5210   TASK: ffff8801396f6aa0  CPU: 0   COMMAND: "bash"  
      MM               PGD          RSS    TOTAL\_VM  
ffff88013975d880  ffff88013a0c5000  1808k   108340k  
     VMA           START       END     FLAGS FILE  
ffff88013a0c4ed0     400000     4d4000 8001875 /bin/bash  
ffff88013cd63210 3804800000 3804820000 8000875 /lib64/ld-2.12.so  
ffff880138cf8ed0 3804c00000 3804c02000 8000075 /lib64/libdl-2.12.so

14. View the Open Files when System Crashed

Execute files command at the crash prompt, which will display the [open files](http://www.thegeekstuff.com/2012/08/lsof-command-examples/) when the system crashed.

**crash> files  
PID: 5210   TASK: ffff8801396f6aa0  CPU: 0   COMMAND: "bash"  
ROOT: /    CWD: /root  
FD       FILE            DENTRY           INODE       TYPE PATH  
 0 ffff88013cf76d40 ffff88013a836480 ffff880139b70d48 CHR  /tty1  
 1 ffff88013c4a5d80 ffff88013c90a440 ffff880135992308 REG  /proc/sysrq-trigger  
255 ffff88013cf76d40 ffff88013a836480 ffff880139b70d48 CHR  /tty1  
..**

15. View System Information when System Crashed

Execute sys command at the crash prompt, which will display system information when the system crashed.

crash> sys  
     KERNEL: /usr/lib/debug/lib/modules/2.6.32-431.5.1.el6.x86\_64/vmlinux  
   DUMPFILE: /var/crash/127.0.0.1-2014-03-26-12:24:39/vmcore  [PARTIAL DUMP]  
       CPUS: 1  
       DATE: Wed Mar 26 12:24:36 2014  
     UPTIME: 00:01:32  
LOAD AVERAGE: 0.17, 0.09, 0.03  
      TASKS: 159  
   NODENAME: elserver1.abc.com  
    RELEASE: 2.6.32-431.5.1.el6.x86\_64  
    VERSION: #1 SMP Fri Jan 10 14:46:43 EST 2014  
    MACHINE: x86\_64  (2132 Mhz)  
     MEMORY: 4 GB  
      PANIC: "Oops: 0002 [#1] SMP " (check log for details)

**Note : For kernel debugging we need following package to be installed,**

**2014 kernel-debuginfo-2.6.32-220.el6.i686.rpm**

**kernel-debuginfo-common-i686-2.6.32-220.el6.i686.rpm**

Listing 2: Panic Routine for NMI Event

**# cat /proc/sys/kernel/unknown\_nmi\_panic  
1  
# sysctl kernel.unknown\_nmi\_panic  
kernel.unknown\_nmi\_panic = 1  
# grep nmi /etc/sysctl.conf  
kernel.unknown\_nmi\_panic = 1**