Linux Crash HOWTO

Norman Patten

nepatten@us.ibm.com

2002−01−30

Revision History

Revision 1.0	2002−01−30	Revised by: NM
Initial release.		

This document describes the installation and usage of the LKCD (Linux Kernel Crash Dump) package.

Table of Contents

1. Introduction

The ***LKCD (Linux Kernel Crash Dump)*** project is a set of kernel patches and utilities to allow a copy of the kernel memory to be saved in the event of a kernel panic. The saved kernel image makes forensics on the kernel panic possible with utilities included in the package. Most commercial Unix operating systems come with similar crash utilities, but this package is fairly new to Linux and has to be added on manually. The LKCD utility is not designed to gather helpful information in the case of a hardware caused panic or a segment violation. The complete LKCD package is available for download at http://lkcd.sourceforge.net/.

1.1. Copyright and License

Send feedback to *nepatten@us.ibm.com*.

---

2. How LKCD Works

When a kernel encounters certain errors it calls the "panic" function which results from a unrecoverable error. This panic results in LKCD initiating a kernel dump where kernel memory is copied out to the pre−designated dump area. The dump device is configured as primary swap by default. The kernel is not completely functional at this point, but there is enough functionality to copy memory to disk. After dump finishes copying memory to disk, the system re−boots. When the system boots back up, it checks for a new crash dump. If a new crash dump is found it is copied from the dump location to the file system, "`/var/log/dump`" directory by default. After copying the image, the system continues to boot normally and forensics can be performed at a later date.

2.1. What You Need

`lkcd-kernelxxx.diff` file for patching the kernel. The kernel version supported will change routinely. `lkcdutils-xx.src.rpm` − this is the utilities source and scripts you will need to setup and read a crash. At the time of this writing there is a i386 binary rpm available from lkcd.sourceforge.net, but you will still need the patches for the startup scripts from the source rpm.

---

3. Installation of lkcd

3.1. Installing From Source Code

Get the `lkcdutils-xxx.src.rpm` and install it using **rpm −i kcdutils−xxx.src.rpm** . This will place a file called `lkcdutils-xxx.tar.gz` in the `/usr/src/redhat/SOURCES` directory. This file is a compressed tar image of the lkcd source tree. Unwind the source in a directory of your choice like "`/usr/src`" with **tar −zxvf kcdutils−xxx.src.rpm** . This will create a directory called "`kcdutils-xxx`" which will contain the LKCD utilities source.

3.2. Building and Installing LKCD Utilities

LKCD used the standard GCC compiler and make files. To build the suite, cd to the LKCD src directory and run **./configure** to build configuration files. The next step is to run **make** to build the utilities, and finally run **make install** to install the utilities and man pages.

3.3. What Gets Installed

```
/etc/sysconfig/dump             # Configuration file for dump
/sbin/lcrash                    # The crash utility
/sbin/lkcd                      # Script to configure and save a crash
/sbin/lkcd_config               # Configuration utility for dump
/sbin/lkcd_ksyms                # Utility for reconstructing kernel symbols
/usr/include/sial_api.h         # Header file for the SIAL API
/usr/lib/libsial.a              # Simple Image Access Language library
/usr/man/man1/lcrash.1          # man page for lcrash
/usr/man/man1/lkcd_config.1     # man page for lkcd_config
/usr/man/man1/lkcd_ksyms.1      # man page for lkcd_ksyms
/usr/share/sial/lcrash/ps.sial  # ps command implementation of SIAL
```

3.4. Installing LKCD Utilities From RPM

You can install the pre−built utilities from rpm by running **rpm −i kcdutils−xxx.rpm** . You will still need to patch the kernel and install the startup script patches. However you can bypass building the utilities step.

3.5. Patching the Kernel

The next step is patching and rebuilding the kernel. You will need to patch the kernel source with the `lkcd-xxx.diff` file you downloaded from http://lkcd.sourceforge.net/. Copy the patch into the same directory as your kernel and run **patch −p0 < lkcd−kernelxxx.diff** . Make sure the patch you use is the same version as the kernel you are patching. Next you will need to configure the kernel to enable crash dump support. By default crash support is turned off after applying the patch. If you use **make menuconfig** or **make xconfig**, the "LKCD support" option is under kernel hacking. You will also need to enable other kernel features you might need. See the The Linux Kernel HOWTO for further details.

3.6. Build and Install the Kernel

The next is building and installing the crash enabled kernel. In the kernel source directory run the following commands in order.

```
make depend
make install
make modules
make modules_install
```

This will build and installing the new kernel, you will also need to copy the `Kerntypes` file from the kernel source to the `/boot` directory. You may also need to edit the `lilo.conf` file to point to you new kernel. See http://www.linuxdoc.org/HOWTO/Kernel–HOWTO.html for more information on building and installing a kernel.

4. Setup, Test, and Running crash

4.1. Setting up crash dump

In order to save a core image that has been written into swap, the image must be saved prior to swap being re−mounted during boot. To accomplish this, the `sysinit` startup file needs to be changed. The lkcd source includes a `scripts` directory which contains patches for various sysinit startup scripts. These patches add the **lkcd config** and **lkcd save** commands to enable crash dumps and to save any existing crash dumps upon startup.

4.2. Testing crash

To force a panic to test you new crash setup, compile the following code with **cc −c −I/usr/src/linux/include panic.c** . After building the `panic.o` module just **insmod panic.o** to panic the kernel.

```
### panic.c ###########################

#define __KERNEL__
# MODULE

# include init_module(void)

int init_module (void)
{
    panic(" panic has been called");
    return 0;
}
```

4.3. Running crash

To view your kernel core file **lcrash** needs to be invoked with a couple of parameters:

```
lcrash [ System.map file ] [ dump image ] [ Kerntypes ]

  Example:

  lcrash /boot/System.map ./dump.1 /boot/Kerntypes
```

It will take a minute to load the kernel image into memory and drop you into the crash shell. At the crash shell prompt you can type a **?** to see the available commands.