Kernel Debugging with netdump and crash

Worcester Linux Users Group

January 13th, 2005

Presenter: Jeff Moyer <jmoyer@redhat.com>

Overview

- Kernel debugging tools
- Kernel crash dump implementations
- Netdump
- crash
- Demo

Kernel debugging tools

- Kernel crash dump tools
  - LKCD
  - Netdump
  - Diskdump
  - Kexec-based dump
- SVR4 “crash” program
  - LKCD (hacked up ancient version of crash)
  - Dave Anderson (the man, the myth, the legend)

Tools (cont'd)

- Kernel debuggers
  - kdb
  - kgdb or gdb stubs
- oops/panic output
- alt-sysrq
- objdump
- printk

Debugger Feature Comparison

	Online	Console	Serial	Network	Post-mortem	Single step
gdb stubs	X		X	X	X	X
kdb	X	X	X			X
crash	X	X	X	X	X	

Crash dump tools

- mcore – ancient, bit rotted
- LKCD – everything and the kitchen sink
- netdump – Red Hat only
- diskdump – Red Hat only
- kexec-based dump – Upstream effort

Netdump

- **Net**work Crash**dump**
- Implemented using the netpoll infrastructure (2.6)
- Requires dedicated netdump server
  - Used to have to be on same network; no more
- Loadable module
  - 2.4 has netconsole.o
  - 2.6 has netconsole.o and netdump.o

Netdump (cont'd)

- 3 bits of functionality
  1. Network crash dump
  2. Network logging
  3. Remote syslog
- 2.4
  - netdump and netlog cannot be configured independently
- 2.6
  - netdump, netlog, and syslog can be configured separately

Netdump: How it works

- Client server
  - Panic()ing system initiates the dump
    - handshake process
  - Server then turns into the client, requesting pages from the panic()ed system
  - client breaks pages up into 1k chunks, due to the default Ethernet MTU of 1500 bytes.
  - At the end of the dump, a sysrq-t is performed

Netdump: supported platforms

- pre RHEL-3 U5
  - x86
- RHEL 3 U5 and beyond (including RHEL 4)
  - x86
  - x86_64
  - ia64
  - ppc64
- netdump server is platform independent.

Dump file format

- ELF core header
  - Can be read by gdb
- ELF header has a NT_TASKSTRUCT note
  - use to squirrel away a pointer to the panic()ing task
- After ELF header, raw dump of memory.

Netdump (in)security

- ssh key shared between client and server
  - used for the distribution of a shared secret, generated upon netdump startup
  - Secret verification only happens one-way.
- UDP unicast used
  - for switched networks, this is generally O.K.

Netdump shortcomings

- No page selection
- No compression
- No encryption
- Takes a long time, and lots of bandwidth

Netdump Setup (server)

- Server
  - rpm -i netdump-server-0.7.4-2.i386.rpm
  - /etc/netdump.conf
    - secure=[01]
  - Set the passwd for the netdump user
  - Optionally, copy scripts from

    `/usr/share/doc/netdump-n-v-r/example_scripts`

    to

    `/var/crash/scripts`
  - service netdump-server start

Netdump Setup (client)

- Client
  - rpm -i netdump-0.7.4-2.i386.rpm
  - modify /etc/sysconfig/netdump
  - service netdump propagate
  - service netdump start

/etc/sysconfig/netdump

```
#LOCALPORT=6666
#DEV=
#NETDUMPADDR=<Required>
#NETDUMPPORT=
#NETDUMPMACADDR=
#IDLETIMEOUT=

#SYSLOGADDR=
#SYSLOGPORT=
#SYSLOGMACADDR=

#NETLOGADDR=
#NETLOGPORT=
#NETLOGMACADDR=
```

Testing your netdump setup

- You will want to enable the magic sysrq key:
  - `# sysctl -w kernel/sysrq=1`
- And panic_on_oops
  - `# sysctl -w kernel/panic_on_oops=1`
- `Check that netlog is working`
  - `# echo h > /proc/sysrq-trigger`
- On the server, you should see a new directory created:
  - `/var/crash/<IPAddr>`
- In that directory will be a file named 'log'
- You can crash the system with:
  - `# echo c > /proc/sysrq-trigger`
- Or by typing alt-sysrq-c

Crash

- Kernel-specific “debugger”
- Can be used on live systems and dump files
- Requires a vmlinux file with debugging symbols
  - Red Hat builds a -debuginfo package with this (though it isn't distributed)
- Knows about kernel specific data structures
  - custom commands
  - can pretty print these structures

Crash (cont'd)

- Supported file formats
  - Any netdump vmcore
  - lkcd up to version 8
  - /dev/kmem (2.4 kernels and upstream 2.6)
  - /dev/crash (Red Hat 2.6 kernels)

Preparing the kernel

- FC-3
  - download the SRPM
    - kernel-2.6.9-1.724_FC3.src.rpm
  - install it
    - rpm -i kernel-2.6.9-1.724_FC3.src.rpm
  - This places the kernel tarball and patches in /usr/src/redhat by default
  - Build the kernel
    - rpmbuild -bb /usr/src/redhat/SPECS/kernel-2.6.spec

prepping kernel (cont'd)

- Install the -debuginfo kernel
  - rpm -i /usr/src/redhat/RPMS/kernel-debuginfo-2.6.9-1.724_FC3.rpm
- And now you're ready to run crash
  - crash /usr/lib/debug/lib/modules/2.6.9-1.724_FC3/vmlinux
- Crash takes arguments for:
  - mapfile (System.map)
  - namelist (vmlinux)
  - dump file (vmcore or /dev/crash)

crash demo

References

- Crash
  - Where to get it:
    - http://people.redhat.com/anderson
    - RHEL or Fedora repositories
  - Documentation
    - http://people.redhat.com/anderson/crash_whitepaper
- Netdump
  - Kernel patches
    - Available as part of the Red Hat kernel SRPMs
  - Documentation
    - http://www.redhat.com/support/wpapers/redhat/netdump/