

Table of Contents

Netdump	1
Support.....	1
Resources / Docs.....	1
Setup at CERN.....	1
SLC5/RHEL5.....	2
Statistics.....	2
analysis example.....	2

Netdump

(netdump is a crash dump over the network, using UDP packets. Has support for specific network hardware only.)

Support

Linux.Support will not blindly debug any kernel crash that arrives on this service. This is highly labour intensive and will be reserved for repeated crashes on recent kernels for production machines, i.e. things that **hurt** and where simple remedies have been tried, including operating system updates. We can forward crash dumps to upstream vendors for machines with a support contract.

Resources / Docs

- <http://www.redhat.com/support/wpapers/redhat/netdump/> : Red Hat whitepaper - ancient but still relevant
- <http://www.redhatmagazine.com/2007/08/15/a-quick-overview-of-linux-kernel-crash-dump-analysis/> : Red Hat magazine introductory article and some links
- <http://people.redhat.com/anderson/> : "crash" utility docs and source, whitepaper, RHEL kernel debuginfo RPMs
- `man crash`: list of commands
- `inside crash: help` command gives a rather detailed help, same as `crash -h` command (outside of the crash application)

Setup at CERN

`lxdump.cern.ch` will accept dumps from any CERN machine, no SSH key authentication for now. The server will notify JanIven, PeterKelemen directly (for now, should probably go via REMEDY eventually). No support yet for central syslogging to this machine. Dumps are collected under `/var/crash/${IP}-${DATE}/`.

Clients will need to have to have the `netdump` RPM installed,

```
NETDUMPADDR=lxdump.cern.ch
NETDUMPPORT=6666
IDLETIMEOUT=60
NETDUMPKEYEXCHANGE=none
```

in `/etc/sysconfig/netdump` and `chkconfig netdump on; service netdump start` (just loads the module).

For Quattor machines, the `ncm-crashdump` component can be used with the following parameters:

```
"/software/components/crashdump/active" = true;
"/software/components/crashdump/client/method" = "netdump";
"/software/components/crashdump/client/remoteserver" = "lxdump.cern.ch";
"/software/components/crashdump/client/remoteport" = 6666;
"/software/components/crashdump/client/idletimeout" = 60;
"/software/components/crashdump/client/keyexchange" = "none";
```

You will still need to explicitly turn on the `netdump` service (e.g via the `chkconfig` NCM component):

```
"/software/components/chkconfig/service/netdump/add" = true;
"/software/components/chkconfig/service/netdump/on" = "345";
```

If your machine is in the IT-CC CDB, simply doing

```
include pro_service_crashdump
```

does all of this. After editing CDB, you will still need to run `ncm-ncd --configure crashdump chkconfig` on the machine.

SLC5/RHEL5

#SLC5

netdump no longer works on SLC5, due to the crash dump mechanism having been changed to kexec (where a "new" kernel actually boots and does the dumping). Crashes end up on local disk under `/var/crash//vmcore`. Presumably one could use the new kernel to transmit the old image to the crash dump server by simply modifying the `kdump` init script. The `netdump` UDP-based protocol seems underdocumented, so one would have to resort to anonymous FTP upload or such.

For a recipe on how to configure `kdump/kexec` for local crash dumping (i.e. save crash dump to local disk), see RedHat's knowledgebase[\[4\]](#).

Statistics

<http://www.kerneloops.org>[\[4\]](#) does statistical analysis on Ooopses (with a small "collector" script, they also use various mailing lists for input), but only just for "recent/upstream" kernels. Could set up private instance at CERN, but server-side PHP code is currently not available (Arjan, 27.05.08: may eventually clean up and publish..)

Until then, we have set up some small sqlite database that tracks crashes by machines/kernel/reason, this is invoked by the `netdump-crash` script and used e.g. when sending the alert mail.

analysis example

(to be continued)

Precondition: the analysis machine needs a `vmlinux` file that corresponds to the one from the crashed machine. Available via the `kernel-debuginfo` RPMS, e.g. from `/afs/cern.ch/project/linux/cern/slc4X/updates/i386/debug/` or Red Hat/Anderson[\[4\]](#)

You also need a `crash` binary that corresponds to the target architecture (i.e 32bit for 32bit-crashes - Bugzilla'ed).

Finding out the exact kernel version for a "vmcore" is left as an exercise for the reader (use the console log or Oops message) - `crash` will not auto-detect this (Bugzilla'ed).

```
# cd /var/crash/128.141.28.38-2006-12-18-12:21
# crash /usr/lib/debug/lib/modules/2.6.9-42.0.2.EL.cern/vmlinux vmcore
.. blabla..
    KERNEL: /usr/lib/debug/lib/modules/2.6.9-42.0.2.EL.cern/vmlinux
    DUMPFILE: vmcore
    CPUS: 1
    DATE: Mon Dec 18 12:21:33 2006
    UPTIME: 00:05:06
    LOAD AVERAGE: 0.83, 0.86, 0.38
    TASKS: 62
    NODENAME: itadc-hp6000
```

NetDump < LinuxSupport < TWiki

```

RELEASE: 2.6.9-42.0.3.EL.cern
VERSION: #1 Fri Oct 6 11:33:33 CEST 2006
MACHINE: i686 (796 Mhz)
MEMORY: 255.9 MB
PANIC: "Oops: 0002 [#1]" (check log for details)
PID: 4192
COMMAND: "bash"
TASK: cb5e3240 [THREAD_INFO: c4e51000]
CPU: 0
STATE: TASK_RUNNING (SYSRQ)

crash> bt
PID: 4192 TASK: cb5e3240 CPU: 0 COMMAND: "bash"
#0 [c4e51df8] netpoll_start_netdump at d0e126ee
#1 [c4e51e18] die at c010683e
#2 [c4e51e48] do_page_fault at c011db74
#3 [c4e51f60] __handle_sysrq at c0242b9d
#4 [c4e51f80] write_sysrq_trigger at c01ae040
#5 [c4e51f88] vfs_write at c016c94e
#6 [c4e51fa4] sys_write at c016ca16
#7 [c4e51fc0] system_call at c0323dd4
EAX: 00000004 EBX: 00000001 ECX: b7d10000 EDX: 00000002
DS: 007b ESI: 00000002 ES: 007b EDI: b7d10000
SS: 007b ESP: bfe1b7f0 EBP: bfe1b810
CS: 0073 EIP: 006917a2 ERR: 00000004 EFLAGS: 00000246

crash> task 4192
PID: 4192 TASK: cb5e3240 CPU: 0 COMMAND: "bash"
struct task_struct {
    state = 0,
    thread_info = 0xc4e51000,
    usage = {
        counter = 9
    },
    flags = 4194560,
    ptrace = 0,
    lock_depth = -1,
    prio = 116,
    static_prio = 120,
    run_list = {
        next = 0xc04569a0,
        prev = 0xc04569a0
    },
    array = 0xc04565e8,
    sleep_avg = 900000000,
    interactive_credit = 101,
    timestamp = 4294906002000000,
    last_ran = 4294906002000000,
    activated = 0,
    [... lots of stuff ...]
crash> ps
  PID   PPID  CPU   TASK      ST  %MEM   VSZ   RSS  COMM
    0      0    0  c037dbe0  RU   0.0     0     0  [swapper]
    1      0    0  cfe0b770  IN   0.2   2212    624  init
[...
 3349      1    0  c1fdcc10  IN   0.0     0     0  [afsd]
[...
 4192   4190    0  cb5e3240  RU   0.6   6180   1688  bash
crash> bt 3349
PID: 3349 TASK: c1fdcc10 CPU: 0 COMMAND: "afsd"
#0 [ce198eb8] schedule at c032149c
#1 [ce198f3c] osi_TimedSleep at d1562a67
#2 [ce198f9c] afs_osi_Wait at d1562412
#3 [ce198fac] afs_Daemon at d1520080
#4 [ce198fd8] afsd_thread at d15687b2
#5 [ce198ff0] kernel_thread_helper at c01041db
crash> mod -s libafs /lib/modules/2.6.9-42.0.3.EL.cern/kernel/fs/openafs/openafs.ko
MODULE  NAME                               SIZE  OBJECT FILE
d158f080 libafs                      540512 /lib/modules/2.6.9-42.0.3.EL.cern/kernel/fs/openafs/openafs.ko

```

```
crash> p afs_linux_lock
afs_linux_lock = $1 =
{int (struct file *, int, struct file_lock *)} 0xd156543c
```

This topic: LinuxSupport > NetDump

Topic revision: r11 - 2009-07-03 - MatthiasSchroeder



Copyright &© 2008-2024 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback