

Shrinking core dumps on the fly

Thomas Gleixner

Linux Plumbers Conference 2012, San Diego





What's wrong with plain core dumps?

- ► Possibly huge storage requirements
- ► Storage of redundant information





Why is that relevant?

- ► Space constraint systems
- ► Low bandwidth to access





Why should we care?

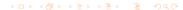
▶ Better debugability on embedded systems





Can we be smarter?

- ► Store only whats relevant
- ▶ Drop information which can be recovered





The access mechanism is there already

/proc/sys/kernel/core_pattern



The filter mechanism

minicore-dumper

- Creates a sparse core file
- Ignore all text sections
- ▶ Store a minimum set of standard information
- ► Allow per executable extra information storage





Host side tools

Debug info generator

- ▶ Extracts debug information for the data to store
- Creates per executable dump config file





Host side tools

Mini core rebuilder

▶ Rebuilds .text sections (executable, libraries)

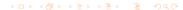




Host side tools

gdb

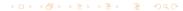
- ► No modifications
- ▶ Just less information accessible





Size

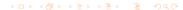
▶ Depends on your dump requirements





Unresolved problems

- ▶ Unnamed map sections (e.g. BSS)
- ► Match some magic gdb expectations





▶ Size reduced from 532M to 800k

```
Program terminated with signal 11, Segmentation fault.
#0 0x0000000000400702 in boom (c=-2) at crash.c:17
17*x = c;
print x
$1 = (int *) 0x0
(gdb) info threads
 4 Thread 0x7f3f33b93700 (LWP 9515) __lll_lock_wait_private () at ../nptl/sysdeps/unix/sysv/linux/x86_6
 3 Thread 0x7f3f34394700 (LWP 9514) 0x00000032b38d350d in write () at ../sysdeps/unix/syscall-template.
 2 Thread 9513 0x00000032b3c07de5 in pthread join (threadid=139909435836160, thread return=0x0) at pthr
* 1 Thread 0x7f3f33392700 (LWP 9516) 0x000000000400702 in boom (c=-2) at crash.c:17
(gdb) bt
#0 0x0000000000400702 in boom (c=-2) at crash.c:17
#1 0x0000000000040072c in bar (b=-1) at crash.c:23
#2 0x00000000000400746 in gluck (a=-3) at crash.c:28
#3 0x000000000040089c in f3 (p=0x0) at crash.c:72
#4 0x00000032b3c06ccb in start_thread (arg=0x7f3f33392700) at pthread_create.c:301
#5 0x00000032b38e0c2d in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:115
(gdb) print tbuf1
$2 = 0x7f3f32691010 ''This is the test text buffer which gets dumped by minicore dumper.''
```





Questions?

