Linux Plumbers Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Modernizing the kdump dump tools

Philipp Rudo
prudo@redhat.com

What's it all about?

Kdump:

- Mechanism for post-mortem (aka. dump) debugging
- Includes kernel & user space tools
- Essential for “service providers”,
  i.e. distros, hardware vendors, etc.

makedumpfile:

- Runs in initrd
- Filter & compress dump

crash:

- Read, parse & display information from dump

What's the problem?

Both tools parse unstable kABI

Both tools are pretty old[citation needed]

Both tools are backward-compatible

From crash's README

- *One size fits all -- the utility can be run on any Linux kernel version dating back to 2.2.5-15. A primary design goal is to always maintain backwards-compatibility.*

Bug in makedumpfile

- Reported: June 2021
- Symptom: Dump corruption on s390
- Problem: mem_section array -> pointer to array (v4.15, Sep 2017)
- Introduced: Workaround for kernel bug in v5.3-v5.5 (Jan 2020)
- Fixed: April 2022, 6 Engineers

Security aspects

- Dump is huge binary file with complex format
- High complexity
  -> high chance for bugs
  -> high chance for security problems
- Especially problematic for customer support

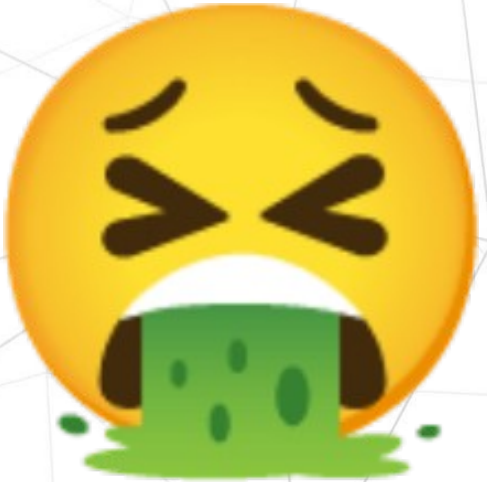
-> Need to reduce complexity

Option 1: Make kABI stable

- Support one version of kABI
- kABI never changes
- All problems are pushed to kernel developers

Option 1: Make kABI stable

- Support one version of kABI
- kABI never changes
- All problems are pushed to kernel developers



Option 2: Trim history

- Support multiple versions of kABI
- Drop support for “old” kernel
- What is “old”?

  -> Either: Lots of work for little to no benefit

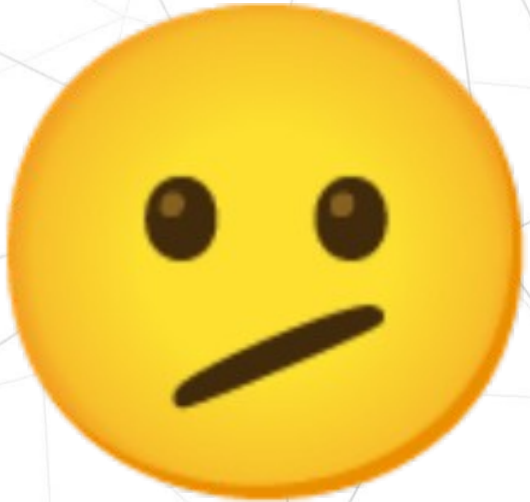
  -> Or: Causing problems to distros

Option 2: Trim history

- Support multiple versions of kABI
- Drop support for “old” kernel
- What is “old”?

  -> Either: Lots of work for little to no benefit

  -> Or: Causing problems to distros

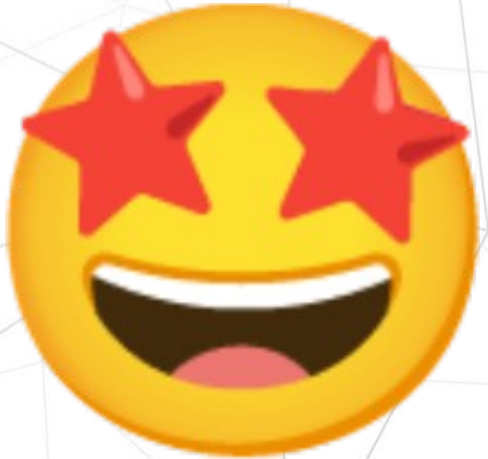


Option 3: Break backward-compatibility

- Support one version of kABI
- Causing problems to distros
- Can move tools to kernel tree
  -> Solves most of the problems

Option 3: Break backward-compatibility

- Support one version of kABI
- Causing problems to distros
- Can move tools to kernel tree
  -> Solves most of the problems



Pros

- Direct mapping between tools and kernel code
  -> Drastically reduced complexity
  -> Easier testing and automation,
  e.g fuzzers, kABI checker
- Well established processes and tools
  in up- & downstream
- Fixes: tag

Cons (upstream)

- New tool(s) maintained in kernel tree
- Additional stable-only patches
- Huge, multi year project
  -> Need to rewrite/redesign crash
- Long transition phase

Cons (downstream)

- New kernel version specific package
- Must update kernel to get tools fix
- Must learn to handle missing features

Thoughts & Opinions?

Linux
Plumbers
Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Thank you!

Linux
Plumbers
Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Backup

Age of Crash

- Git Jan 2014 (crash-7.0.4)
- Mailing list Oct 2005
- ChangeLog Apr 2004 (crash-3.7-5.4)
- Copyright statement earliest 1999
- LKCD 1.0 Nov 1999
- Release 2.2.5 March 1999
- GDB 5.0 May 2000

Background:

k: 83e3c48729d9 ("mm/sparsemem: Allocate mem_section at runtime for CONFIG_SPARSEMEM_EXTREME=y")
- mem_section array -> pointer to array ,Sep 2017, 4.15

k: a0b1280368d1 ("kdump: write correct address of mem_section into vmcoreinfo"), Jan 2018, v4.15
- “revert” type change in vmcoreinfo DWARF in vmlinux

m: 14876c4 ("[PATCH makedumpfile] handle mem_section as either a pointer or an array"), Feb 2018
- Strategy:
  - parse mem_section assuming it’s an array
  - if SPARSEMEM_EXTREME retry assuming mem_section is pointer to array
  - hope one failed

k = kernel, m = makedumpfile

Bug:

m: e113f1c ("[PATCH] cope with not-present mem section"), Jan 2020

- workaround for kernel bug present in v5.3 - v5.5
- validation always succeeds on s390

m: 81b79c5 ("[PATCH] Avoid false-positive failure in mem_seciton validation"), Feb 2020

- only retry when first validation failed

  -> dump corruption on s390, with -x option

m: 6d0d95e ("[PATCH] Avoid false-positive mem_section validation with vmlinux"), Apr 2022

- final fix (hopefully)

k = kernel, m = makedumpfile

Alternatives to crash

- /scripts/gdb
- crash-python
- drgn

Tools to be included

- crash
- makedumpfile
- vmcore-dmesg (kexec-tools)
- vmcore-uname (new)