https://lore.kernel.org/patchwork/patch/470228/ was used as a reference.

scripts/decode_stacktrace.sh is used to understand the stack dump outputs.
Quoting Sasha Levin <sasha.levin@oracle.com> , when we read the stack dumps as they are, they look something like this.

```
[    6.906437]  [<ffffffff811f0e90>] ? backtrace_test_irq_callback+0x20/0x20
[    6.907121]  [<ffffffff84388ce8>] dump_stack+0x52/0x7f
[    6.907640]  [<ffffffff811f0ec8>] backtrace_regression_test+0x38/0x110
[    6.908281]  [<ffffffff813596a0>] ? proc_create_data+0xa0/0xd0
[    6.908870]  [<ffffffff870a8040>] ? proc_modules_init+0x22/0x22
[    6.909480]  [<ffffffff810020c2>] do_one_initcall+0xc2/0x1e0
[...]
```

However, most of the text we get is pure garbage.
Even I, a kernel newbie can see that the only thing useful in the above text is the function name.
“The kernel address and the offset into the function are not really useful as we use various kernel code versions and configurations. “ Sasha said.

After looking at a few issues reported to the kernel community (before 2015),
I noticed that they send each other these stack dumps and end up asking for 1 or 2 more lines from ‘addr2line’ output.
Example:
- https://lkml.org/lkml/2014/6/4/278
- https://lkml.org/lkml/2014/5/7/844
- https://lkml.org/lkml/2014/5/12/355

^--These pointers have been taken directly from
https://lore.kernel.org/patchwork/patch/470228/ .

This seems to slow down the entire debugging process and thus the script decode_stacktrace.sh is
an attempt to convert all kernel addresses into line numbers.
Which means after `scripts/decode_stacktrace.sh -r `uname -r` < input.log > output.log`, the stack dump will look something like this.

```
[  635.148361]  dump_stack (lib/dump_stack.c:52)
[  635.149127]  warn_slowpath_common (kernel/panic.c:418)
```

[635.150214] warn_slowpath_null (kernel/panic.c:453)
[635.151031] _oalloc_pages_slowpath+0x6a/0x7d0
[635.152171] ? zone_watermark_ok (mm/page_alloc.c:1728)
[635.152988] ? get_page_from_freelist (mm/page_alloc.c:1939)
[635.154766] __alloc_pages_nodemask (mm/page_alloc.c:2766)

This seems way simpler than earlier because it is more human readable.
The dump output sent along with the task
https://syzkaller.appspot.com/text?tag=CrashLog&x=17f5743fe00000
ends up turning into the following text.

[[0;32m OK [0m] Reached target Login Prompts.
[[0;32m OK [0m] Reached target Multi-User System.
[[0;32m OK [0m] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[[0;32m OK [0m] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 9 syzkaller ttyS0

Warning: Permanently added '10.128.10.30' (ECDSA) to the list of known hosts.
executing program
executing program
syzkaller login: [844.569463][T7073]
==
[844.569638][T7073] BUG: KASAN: null-ptr-deref in
uncore_pmu_event_add+0xa25/0x16d0
[844.569649][T7073] Read of size 8 at addr 0000000000000000 by task
syz-executor764/7073
[844.569653][T7073]
[844.569668][T7073] CPU: 0 PID: 7073 Comm: syz-executor764 Not tainted
5.7.0-rc1-next-20200415-syzkaller #0
[844.569676][T7073] Hardware name: Google Google Compute Engine/Google
Compute Engine, BIOS Google 01/01/2011
[844.569701][T7073] Call Trace:
[844.569734][T7073] <IRQ>
[844.569805][T7073] dump_stack (lib/earlycpio.c:72)
[844.569823][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.569884][T7073] __kasan_report.cold+0x5/0x4d

[844.569902][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.569919][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.569931][T7073] kasan_report (mm/kasan/report.c:462
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/report.c:529)
[844.569953][T7073] check_memory_region (mm/kasan/generic.c:113
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:134
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:165
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:183
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:192)
[844.569969][T7073] uncore_pmu_event_add+0xa25/0x16d0
[844.570036][T7073] ? interrupt_entry+0xb8/0xc0
[844.570050][T7073] ? apic_timer_interrupt+0xa/0x20
[844.570056][T7073] </IRQ>
[844.570104][T7073] ? vga16fb_fillrect+0x9da/0x1960
[844.570118][T7073] ? vga16fb_fillrect+0x9c6/0x1960
[844.570161][T7073] ? __sanitizer_cov_trace_pc+0x31/0x60
[844.570176][T7073] ? vga16fb_fillrect+0x9c6/0x1960
[844.570192][T7073] ? memcpy (mm/kasan/common.c:105 (discriminator 1))
[844.570229][T7073] ? bit_clear_margins (drivers/video/fbdev/core/fbcon.h:130
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/bitblit.c:67)
[844.570244][T7073] ? bit_bmove (drivers/video/fbdev/core/bitblit.c:211)
[844.570259][T7073] ? vesafb_probe.cold+0x1162/0x1162
[844.570301][T7073] ? fbcon_clear_margins (drivers/video/fbdev/core/fbcon.c:1339)
[844.570317][T7073] ? fbcon_switch (drivers/video/fbdev/core/fbcon.c:696)
[844.570337][T7073] ? fbcon_set_def_font (drivers/video/fbdev/core/fbcon.c:2283)
[844.570365][T7073] ? fbcon_cursor (drivers/video/fbdev/core/fbcon.c:306
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbcon.c:2413)
[844.570377][T7073] ? bit_clear (drivers/video/fbdev/core/bitblit.c:177)
[844.570412][T7073] ? is_console_locked (kernel/printk/printk.c:2342)
[844.570424][T7073] ? fbcon_set_origin (drivers/video/fbdev/core/fbcon.c:1385)
[844.570467][T7073] ? redraw_screen (drivers/tty/vt/vt.c:1042)
[844.570479][T7073] ? vesafb_probe.cold+0x1162/0x1162
[844.570494][T7073] ? respond_string+0x290/0x290
[844.570510][T7073] ? fbcon_set_palette (drivers/video/fbdev/core/fbcon.c:600)
[844.570527][T7073] ? fbcon_modechanged (drivers/video/fbdev/core/fbcon.c:264
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbcon.c:3462)
[844.570545][T7073] ? fbcon_set_all_vcs (drivers/video/fbdev/core/fbcon.c:3432)
[844.570564][T7073] ? fbcon_update_vcs (drivers/video/fbdev/core/fbcon.c:2196)
[844.570578][T7073] ? fb_set_var (drivers/video/fbdev/core/fbmem.c:1095)

[844.570593][T7073] ? fb_blank (drivers/video/fbdev/core/fbmem.c:959)
[844.570617][T7073] ? mark_held_locks+0xe0/0xe0
[844.570666][T7073] ? _raw_spin_unlock_irqrestore (kernel/locking/spinlock.c:174)
[844.570685][T7073] ? do_fb_ioctl (drivers/video/fbdev/core/fbmem.c:1161)
[844.570700][T7073] ? fb_mmap (./include/linux/fb.h:637
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbmem.c:1142)
[844.570786][T7073] ? tomoyo_execute_permission (security/tomoyo/file.c:755)
[844.570818][T7073] ? __sanitizer_cov_trace_switch+0x45/0x70
[844.570878][T7073] ? do_vfs_ioctl (fs/ioctl.c:640
/home/bkkarthik/Workspace/eudyptula/linux-next/fs/ioctl.c:721)
[844.570908][T7073] ? up_read (kernel/locking/rwsem.c:1592)
[844.570924][T7073] ? down_read_nested+0x430/0x430
[844.570948][T7073] ? fb_ioctl (drivers/video/fbdev/core/fbmem.c:757
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbmem.c:831)
[844.570961][T7073] ? do_fb_ioctl (drivers/video/fbdev/core/fbmem.c:873)
[844.570974][T7073] ? ksys_ioctl (fs/ioctl.c:760)
[844.570991][T7073] ? __x64_sys_ioctl (fs/ioctl.c:760)
[844.571004][T7073] ? lockdep_hardirqs_on+0x463/0x620
[844.571041][T7073] ? do_syscall_64 (arch/x86/entry/common.c:468)
[844.571058][T7073] ? entry_SYSCALL_64_after_hwframe
(arch/x86/entry/entry_64.S:126)
[844.571079][T7073]
==========
[844.571083][T7073] Disabling lock debugging due to kernel taint
[844.571090][T7073] Kernel panic - not syncing: panic_on_warn set ...
[844.571103][T7073] CPU: 0 PID: 7073 Comm: syz-executor764 Tainted: G B
5.7.0-rc1-next-20200415-syzkaller #0
[844.571110][T7073] Hardware name: Google Google Compute Engine/Google
Compute Engine, BIOS Google 01/01/2011
[844.571113][T7073] Call Trace:
[844.571117][T7073] <IRQ>
[844.571128][T7073] dump_stack (lib/earlycpio.c:72)
[844.571176][T7073] panic (kernel/panic.c:324 (discriminator 5))
[844.571189][T7073] ? add_taint.cold+0x16/0x16
[844.571205][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.571223][T7073] ? trace_hardirqs_off+0x50/0x220
[844.571234][T7073] ? trace_hardirqs_off+0x50/0x220
[844.571248][T7073] ? uncore_pmu_event_add+0xa25/0x16d0

[844.571260][T7073] end_report (mm/kasan/report.c:107)
[844.571272][T7073] __kasan_report.cold+0xd/0x4d
[844.571287][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.571300][T7073] ? uncore_pmu_event_add+0xa25/0x16d0
[844.571311][T7073] kasan_report (mm/kasan/report.c:462
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/report.c:529)
[844.571324][T7073] check_memory_region (mm/kasan/generic.c:113
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:134
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:165
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:183
/home/bkkarthik/Workspace/eudyptula/linux-next/mm/kasan/generic.c:192)
[844.571338][T7073] uncore_pmu_event_add+0xa25/0x16d0
[844.571348][T7073] ? interrupt_entry+0xb8/0xc0
[844.571360][T7073] ? apic_timer_interrupt+0xa/0x20
[844.571364][T7073] </IRQ>
[844.571377][T7073] ? vga16fb_fillrect+0x9da/0x1960
[844.571386][T7073] ? vga16fb_fillrect+0x9c6/0x1960
[844.571396][T7073] ? __sanitizer_cov_trace_pc+0x31/0x60
[844.571408][T7073] ? vga16fb_fillrect+0x9c6/0x1960
[844.571419][T7073] ? memcpy (mm/kasan/common.c:105 (discriminator 1))
[844.571432][T7073] ? bit_clear_margins (drivers/video/fbdev/core/fbcon.h:130
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/bitblit.c:67)
[844.571443][T7073] ? bit_bmove (drivers/video/fbdev/core/bitblit.c:211)
[844.571454][T7073] ? vesafb_probe.cold+0x1162/0x1162
[844.571471][T7073] ? fbcon_clear_margins (drivers/video/fbdev/core/fbcon.c:1339)
[844.571483][T7073] ? fbcon_switch (drivers/video/fbdev/core/fbcon.c:696)
[844.571497][T7073] ? fbcon_set_def_font (drivers/video/fbdev/core/fbcon.c:2283)
[844.571513][T7073] ? fbcon_cursor (drivers/video/fbdev/core/fbcon.c:306
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbcon.c:2413)
[844.571523][T7073] ? bit_clear (drivers/video/fbdev/core/bitblit.c:177)
[844.571534][T7073] ? is_console_locked (kernel/printk/printk.c:2342)
[844.571544][T7073] ? fbcon_set_origin (drivers/video/fbdev/core/fbcon.c:1385)
[844.571557][T7073] ? redraw_screen (drivers/tty/vt/vt.c:1042)
[844.571567][T7073] ? vesafb_probe.cold+0x1162/0x1162
[844.571580][T7073] ? respond_string+0x290/0x290
[844.571592][T7073] ? fbcon_set_palette (drivers/video/fbdev/core/fbcon.c:600)
[844.571604][T7073] ? fbcon_modechanged (drivers/video/fbdev/core/fbcon.c:264
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbcon.c:3462)
[844.571617][T7073] ? fbcon_set_all_vcs (drivers/video/fbdev/core/fbcon.c:3432)

[844.571630][T7073] ? fbcon_update_vcs (drivers/video/fbdev/core/fbcon.c:2196)
[844.571642][T7073] ? fb_set_var (drivers/video/fbdev/core/fbmem.c:1095)
[844.571654][T7073] ? fb_blank (drivers/video/fbdev/core/fbmem.c:959)
[844.571667][T7073] ? mark_held_locks+0xe0/0xe0
[844.571694][T7073] ? _raw_spin_unlock_irqrestore (kernel/locking/spinlock.c:174)
[844.571708][T7073] ? do_fb_ioctl (drivers/video/fbdev/core/fbmem.c:1161)
[844.571719][T7073] ? fb_mmap (./include/linux/fb.h:637
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbmem.c:1142)
[844.571732][T7073] ? tomoyo_execute_permission (security/tomoyo/file.c:755)
[844.571751][T7073] ? __sanitizer_cov_trace_switch+0x45/0x70
[844.571763][T7073] ? do_vfs_ioctl (fs/ioctl.c:640
/home/bkkarthik/Workspace/eudyptula/linux-next/fs/ioctl.c:721)
[844.571781][T7073] ? up_read (kernel/locking/rwsem.c:1592)
[844.571793][T7073] ? down_read_nested+0x430/0x430
[844.571806][T7073] ? fb_ioctl (drivers/video/fbdev/core/fbmem.c:757
/home/bkkarthik/Workspace/eudyptula/linux-next/drivers/video/fbdev/core/fbmem.c:831)
[844.571818][T7073] ? do_fb_ioctl (drivers/video/fbdev/core/fbmem.c:873)
[844.571829][T7073] ? ksys_ioctl (fs/ioctl.c:760)
[844.571843][T7073] ? __x64_sys_ioctl (fs/ioctl.c:760)
[844.571854][T7073] ? lockdep_hardirqs_on+0x463/0x620
[844.571867][T7073] ? do_syscall_64 (arch/x86/entry/common.c:468)
[844.571880][T7073] ? entry_SYSCALL_64_after_hwframe
(arch/x86/entry/entry_64.S:126)
[845.748127][T7073] Shutting down cpus with NMI
[845.750244][T7073] Kernel Offset: disabled
[846.436980][T7073] Rebooting in 86400 seconds..

I did not read the log completely, but I was able to make out the following by reading the first few lines, I was able to notice that it is way simpler to read logs after running them through decode_stacktrace.sh because it converts kernel addresses into line numbers making the logs more human friendly to read. I was able to understand the following from the logs.

reached runlevel 4
reached runlevel 5
hostname -- Debian GNU/Linux 9 syzkaller

login ttyS0 -- syzkaller
[BUG] KASAN: null pointer dereferenced in uncore_pmu_event_add+0xa25/0x16d0
Kernel Version: 5.7.0-rc1-next-20200415-syzkaller #0
Physical System: Google Compute Engine with BIOS dated 01/01/2011
start trace:
<IRQ>
dump_stack using lib/earlycpio.c:72
add new uncore pmu event 0xa25/0x16d0
KASAN report at 0x5/0x4d
add new uncore pmu event 0xa25/0x16d0
add new uncore pmu event 0xa25/0x16d0
KASAN report using mm/kasan/report.c
check memory region using mm/kasan/generic.c
add new uncore pmu event 0xa25/0x16d0
interrupt entry 0xb8/0xc0
apic timer interrupt 0xa/0x20
</IRQ>

..
Video drivers doing their job
..

KASAN sanitize 0x31/0x60
memory copy using mm/kasan/common.c -- discriminator 1 (I do not have a good understanding of this.)
..
video drivers doing their job
..

...

I was able to notice that there is a bug due to dereferencing a null pointer
(I could be wrong, but this seems pretty obvious after reading the output obtainer from decode_stacktrace.sh)
thus task syz-executer764/7073 [T7073] ended up with read operation of size 8 at 0000000000000000 which in turn results
in the disabling of lock debugging (noticed as a kernel taint) and ends up with a kernel panic_on_warn being set (bit being set) due to “not syncing” and the kernel decides to reach runlevel 6 (reboot.target).

(please excuse me if the terminologies have been used wrongly).
I have probably not understood the situation completely, but it seems to me that dereferencing a null
pointer ended up in reaching runlevel 6, and probably a boot loop.

I can tell I now have an idea of how to use decode_stacktrace.sh to look for bugs (well, at least the most obvious ones).