

Title: Decoding Syzkaller Bug Stack Trace using decode_stacktrace.sh

Steps:

- Selected bug:

<https://syzkaller.appspot.com/bug?id=b97ec15bfe317ac1ddccb41f2a913d4f7a31c6d7>

- Saved stack trace into `stacktrace.txt`

- Built kernel with debug info (`CONFIG_DEBUG_INFO=y`)

- Installed required packages (`libelf-dev`)

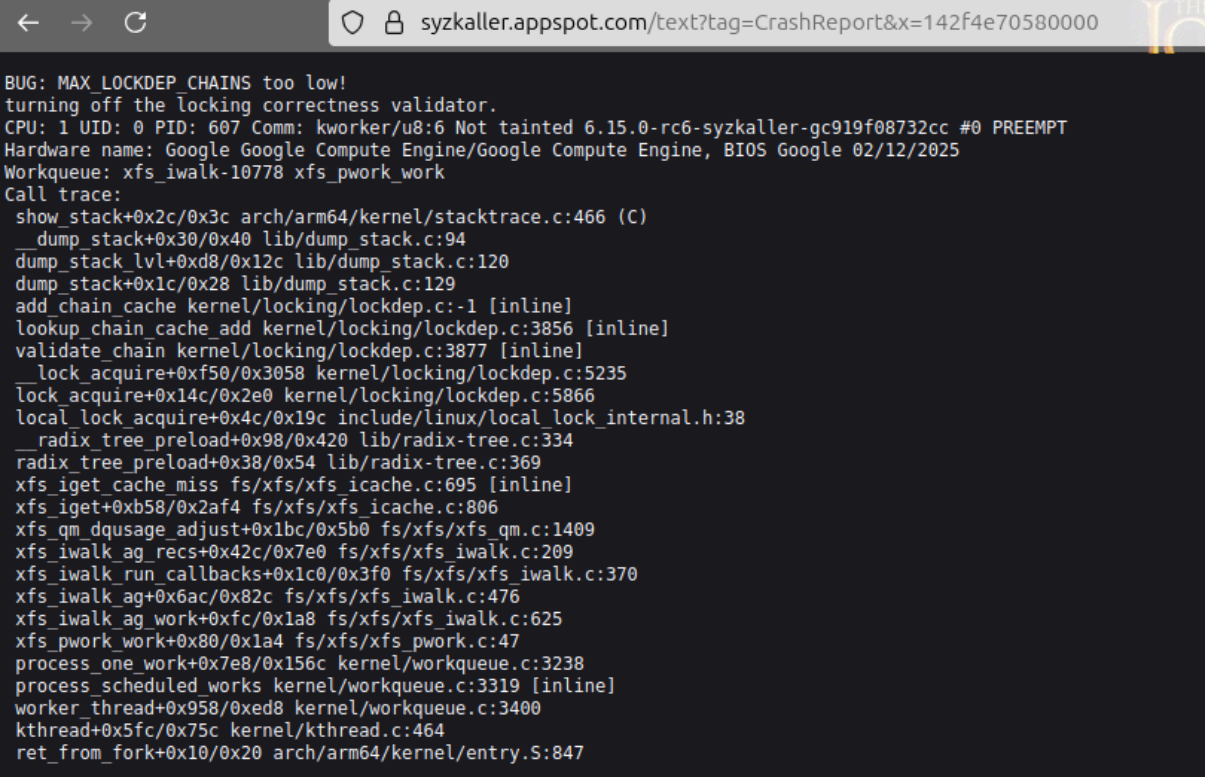
- Ran:

```
./scripts/decode_stacktrace.sh vmlinux < stacktrace.txt > decoded_trace.txt
```

Output clearly mapped addresses to function:line references in kernel code.

Regards,

Arnav Kapoor



The screenshot shows a web browser window with the address bar displaying `syzkaller.appspot.com/text?tag=CrashReport&x=142f4e70580000`. The main content area shows a bug report for "BUG: MAX_LOCKDEP CHAINS too low!". The report includes details about the CPU, UID, PID, and the hardware name. The stack trace is displayed as a list of function names and addresses, with the top of the trace being `show_stack+0x2c/0x3c arch/arm64/kernel/stacktrace.c:466 (C)`. The stack trace is formatted with indentation to show the call chain.

```
BUG: MAX_LOCKDEP CHAINS too low!
turning off the locking correctness validator.
CPU: 1 UID: 0 PID: 607 Comm: kworker/u8:6 Not tainted 6.15.0-rc6-syzkaller-gc919f08732cc #0 PREEMPT
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025
Workqueue: xfs_iwalk-10778 xfs_pwork_work
Call trace:
 show_stack+0x2c/0x3c arch/arm64/kernel/stacktrace.c:466 (C)
  dump_stack+0x30/0x40 lib/dump_stack.c:94
 dump_stack_lvl+0xd8/0x12c lib/dump_stack.c:120
 dump_stack+0x1c/0x28 lib/dump_stack.c:129
 add_chain_cache kernel/locking/lockdep.c:-1 [inline]
 lookup_chain_cache_add kernel/locking/lockdep.c:3856 [inline]
 validate_chain kernel/locking/lockdep.c:3877 [inline]
 __lock_acquire+0xf50/0x3058 kernel/locking/lockdep.c:5235
 lock_acquire+0x14c/0x2e0 kernel/locking/lockdep.c:5866
 local_lock_acquire+0x4c/0x19c include/linux/local_lock_internal.h:38
 __radix_tree_preload+0x98/0x420 lib/radix-tree.c:334
 radix_tree_preload+0x38/0x54 lib/radix-tree.c:369
 xfs_iget_cache_miss fs/xfs/xfs_iget.c:695 [inline]
 xfs_iget+0xb58/0x2af4 fs/xfs/xfs_iget.c:806
 xfs_qm_dqusage_adjust+0x1bc/0x5b0 fs/xfs/xfs_qm.c:1409
 xfs_iwalk_ag_recs+0x42c/0x7e0 fs/xfs/xfs_iwalk.c:209
 xfs_iwalk_run_callbacks+0x1c0/0x3f0 fs/xfs/xfs_iwalk.c:370
 xfs_iwalk_ag+0x6ac/0x82c fs/xfs/xfs_iwalk.c:476
 xfs_iwalk_ag_work+0xfc/0x1a8 fs/xfs/xfs_iwalk.c:625
 xfs_pwork_work+0x80/0x1a4 fs/xfs/xfs_pwork.c:47
 process_one_work+0x7e8/0x156c kernel/workqueue.c:3238
 process_scheduled_works kernel/workqueue.c:3319 [inline]
 worker_thread+0x958/0xed8 kernel/workqueue.c:3400
 kthread+0x5fc/0x75c kernel/kthread.c:464
 ret_from_fork+0x10/0x20 arch/arm64/kernel/entry.S:847
```