Contribution ID: **233** Type: **not specified**

BPF Debugging

Wednesday, September 11, 2019 3:23 PM (22 minutes)

Debugging BPF program logic is hard these days.
Developers typically write their programs and
then checking map values or perf_event outputs
make sense or not. For tricky issues, temporary
maps or bpf_trace_printk are used so developer
can get more insight about what happens. But
this requires possibly multiple rounds of
modifying sources, recompilation and redeployment, etc.

This discussion surrounds creating bpf debugging
tool, bdb (bpf debugger) similar naming after gdb/lldb.
This tool should try to do what gdb for ELF execution.
- specify breakpoints at source/xlated/jitted level
- retrieve data for registers, stacks and globals(maps)
and presented at both register and variable level.
- different conditions to retrieve data, e.g.,
running 100 times, only if this variable == 1.
this will require kernel to live patch bpf codes.
- modifying data (register, stack slot, globals)?
how does this interact with verifier to ensure safety.
- this will leverage BTF and existing test_run framework.
- production debugging vs. qemu debugging
qemu debugging may be truely single-step.

I agree to abide by the anti-harassment policy

Yes

Primary author: SONG, Yonghong

Presenter: SONG, Yonghong

Session Classification: BPF MC