shaping tomorrow with you

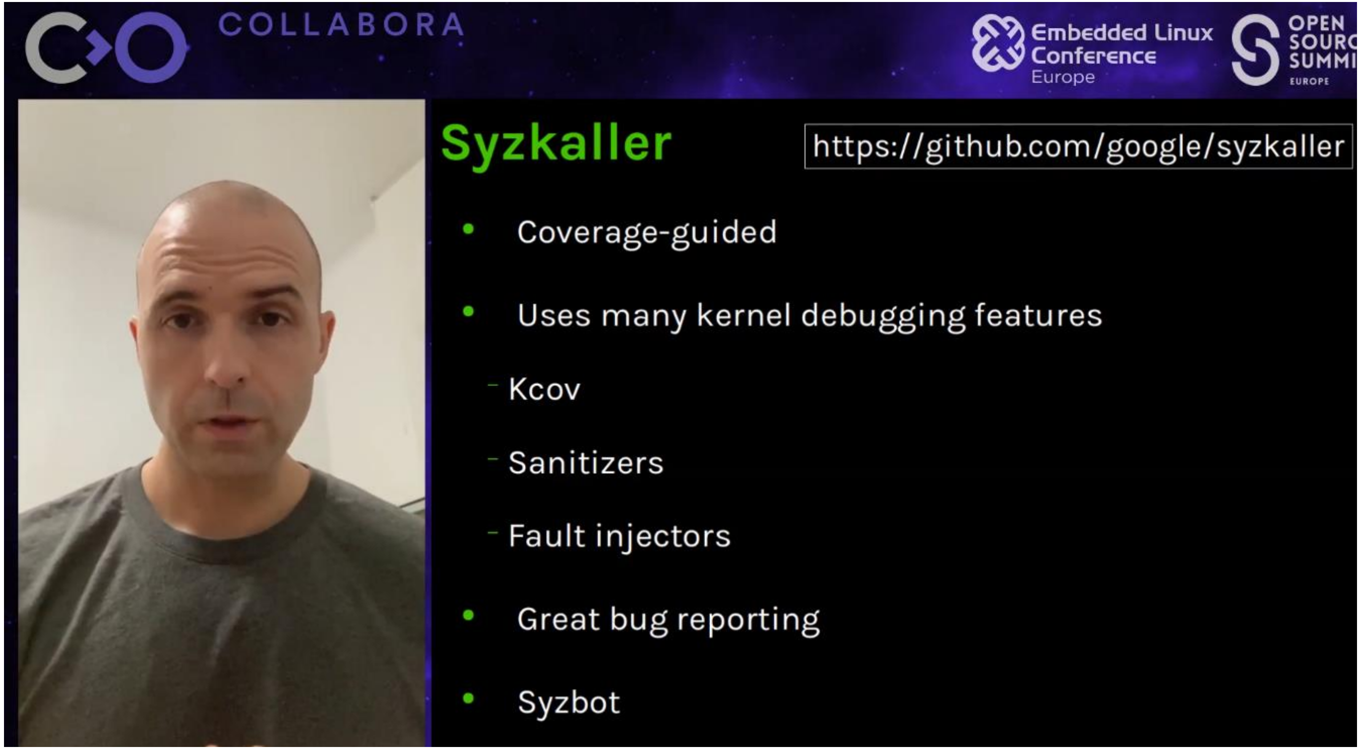
Open Source Summit Europe 2020 Colalboraの発表抜粋

株式会社富士通コンピュータテクノロジーズ
信田 圭哉

Collaboraの発表抜粋

- Fuzzing Linux Drivers with Syzkaller
  - https://osseu2020.sched.com/event/eCEp/fuzzing-linux-drivers-with-syzkaller-ricardo-canuelo-navarro-collabora?iframe=no&w=100%&sidebar=yes&bg=no
  - Linuxドライバのファジングを行うツールの紹介
- youtube
  - https://www.youtube.com/watch?v=REQcQSOIX9U&list=PLZjq3una5SrDeo4RM5UZyZTisSuLd_3gb&index=3
- その他、Collaboraの発表一覧
  - https://www.collabora.com/news-and-blog/news-and-events/open-source-summit-europe-elce-2020.html

Fuzzing Linux Driver with Syzkaller抜粋

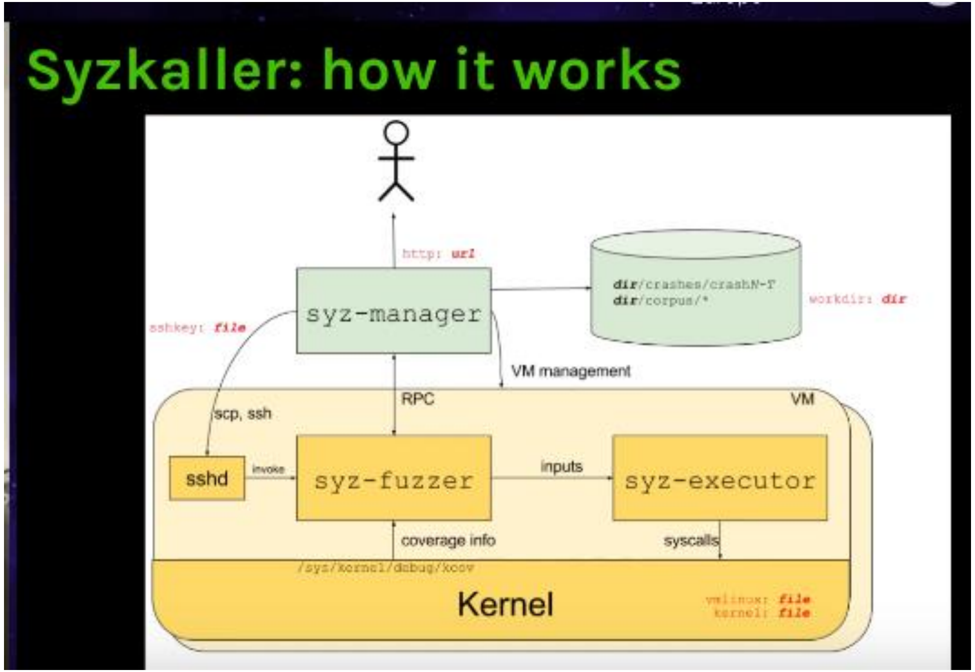


Syzkaller

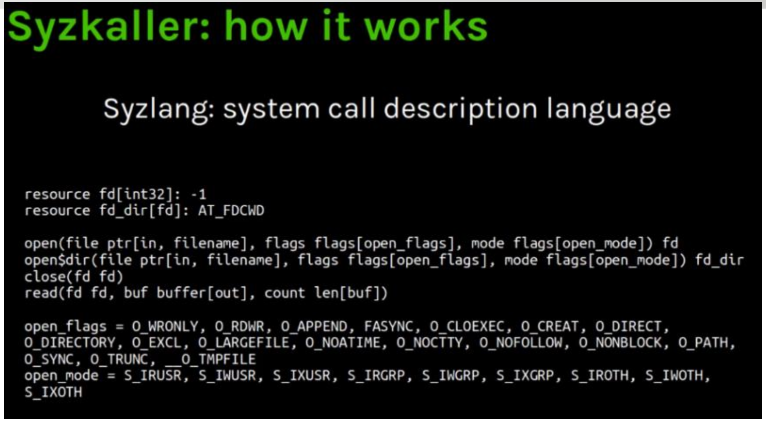
https://github.com/google/syzkaller

- Coverage-guided
- Uses many kernel debugging features
  - Kcov
  - Sanitizers
  - Fault injectors
- Great bug reporting
- Syzbot

Fuzzing Linux Driver with Syzkaller抜粋



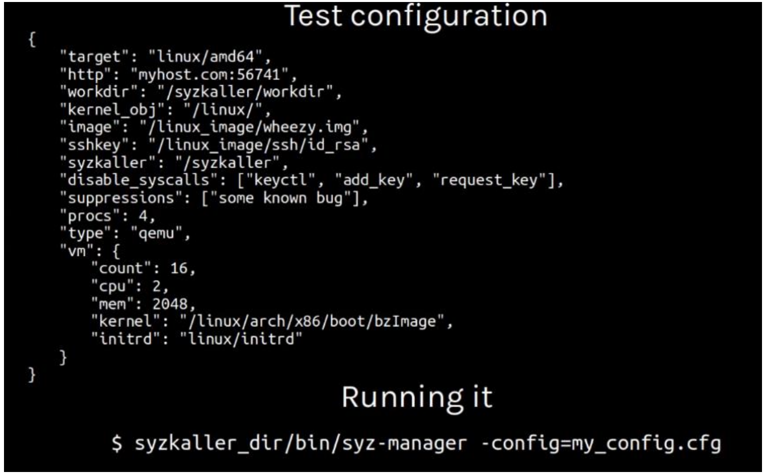
Syzkaller: how it works



Syzkaller: how it works

Syzlang: system call description language

```
resource fd[int32]: -1
resource fd_dir[fd]: AT_FDCWD

open(file ptr[in, filename], flags flags[open_flags], mode flags[open_mode]) fd
open$dir(file ptr[in, filename], flags flags[open_flags], mode flags[open_mode]) fd_dir
close(fd fd)
read(fd fd, buf buffer[out], count len[buf])

open_flags = O_WRONLY, O_RDWR, O_APPEND, FASYNC, O_CLOEXEC, O_CREAT, O_DIRECT,
O_DIRECTORY, O_EXCL, O_LARGEFILE, O_NOATIME, O_NOCTTY, O_NOFOLLOW, O_NONBLOCK, O_PATH,
O_SYNC, O_TRUNC, __O_TMPFILE
open_mode = S_IRUSR, S_IWUSR, S_IXUSR, S_IRGRP, S_IWGRP, S_IXGRP, S_IROTH, S_IWOTH,
S_IXOTH
```



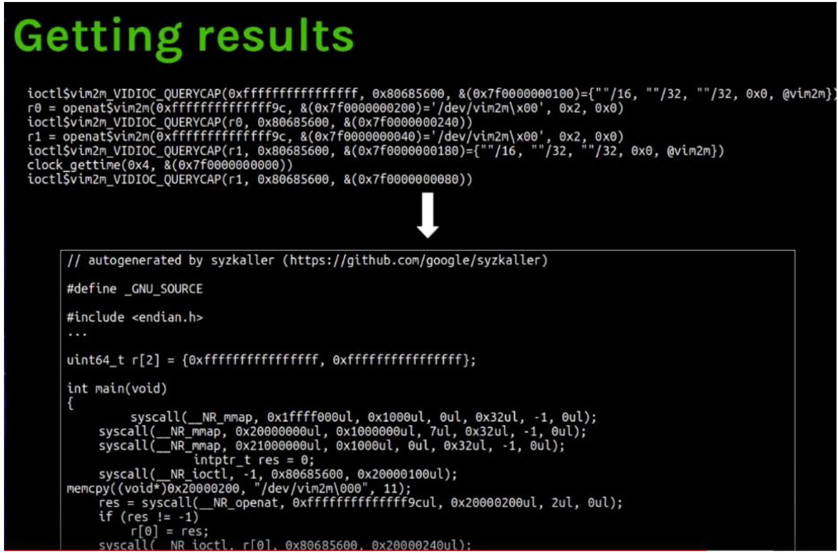
Test configuration

```
{
    "target": "linux/amd64",
    "http": "myhost.com:56741",
    "workdir": "/syzkaller/workdir",
    "kernel_obj": "/linux/",
    "image": "/linux_image/wheezy.img",
    "sshkey": "/linux_image/ssh/id_rsa",
    "syzkaller": "/syzkaller",
    "disable_syscalls": ["keyctl", "add_key", "request_key"],
    "suppressions": ["some known bug"],
    "procs": 4,
    "type": "qemu",
    "vm": {
        "count": 16,
        "cpu": 2,
        "mem": 2048,
        "kernel": "/linux/arch/x86/boot/bzImage",
        "initrd": "linux/initrd"
    }
}
```

Running it

```
$ syzkaller_dir/bin/syz-manager -config=my_config.cfg
```

Fuzzing Linux Driver with Syzkaller抜粋



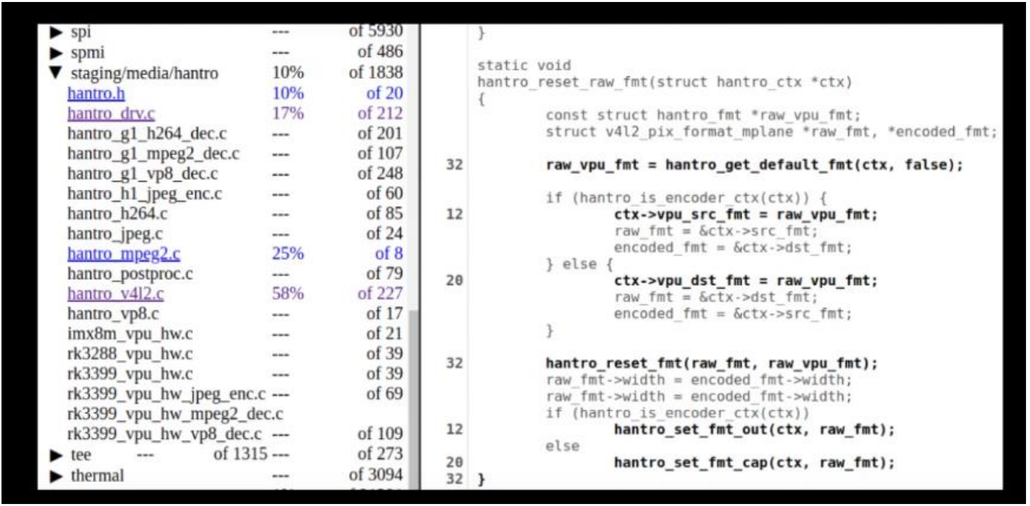
Getting results

```
ioctl$vim2m_VIDIOC_QUERYCAP(0xffffffffffffffff, 0x80685600, &(0x7f0000000100)={""/16, ""/32, ""/32, 0x0, @vim2m}
r0 = openat$vim2m(0xffffffffffffff9c, &(0x7f0000000200)='/dev/vim2m\x00', 0x2, 0x0)
ioctl$vim2m_VIDIOC_QUERYCAP(r0, 0x80685600, &(0x7f0000000240))
r1 = openat$vim2m(0xffffffffffffff9c, &(0x7f0000000040)='/dev/vim2m\x00', 0x2, 0x0)
ioctl$vim2m_VIDIOC_QUERYCAP(r1, 0x80685600, &(0x7f0000000180)={""/16, ""/32, ""/32, 0x0, @vim2m})
clock_gettime(0x4, &(0x7f0000000000))
ioctl$vim2m_VIDIOC_QUERYCAP(r1, 0x80685600, &(0x7f0000000080))
```

⬇

```
// autogenerated by syzkaller (https://github.com/google/syzkaller)

#define _GNU_SOURCE

#include <endian.h>
...

uint64_t r[2] = {0xffffffffffffffff, 0xffffffffffffffff};

int main(void)
{
            syscall(__NR_mmap, 0x1ffff000ul, 0x1000ul, 0ul, 0x32ul, -1, 0ul);
      syscall(__NR_mmap, 0x20000000ul, 0x1000000ul, 7ul, 0x32ul, -1, 0ul);
      syscall(__NR_mmap, 0x21000000ul, 0x1000ul, 0ul, 0x32ul, -1, 0ul);
                  intptr_t res = 0;
      syscall(__NR_ioctl, -1, 0x80685600, 0x20000100ul);
memcpy((void*)0x20000200, "/dev/vim2m\000", 11);
      res = syscall(__NR_openat, 0xffffffffffffff9cul, 0x20000200ul, 2ul, 0ul);
      if (res != -1)
            r[0] = res;
      syscall(__NR_ioctl, r[0], 0x80685600, 0x20000240ul);
```

Syzbot:
https://syzkaller.appspot.com/upstream



```
▶ spi                             ---     of 5930
▶ spmi                            ---     of 486
▼ staging/media/hantro            10%     of 1838
  hantro.h                        10%     of 20
  hantro_drv.c                    17%     of 212
  hantro_g1_h264_dec.c            ---     of 201
  hantro_g1_mpeg2_dec.c           ---     of 107
  hantro_g1_vp8_dec.c             ---     of 248
  hantro_h1_jpeg_enc.c            ---     of 60
  hantro_h264.c                   ---     of 85
  hantro_jpeg.c                   ---     of 24
  hantro_mpeg2.c                  25%     of 8
  hantro_postproc.c               ---     of 79
  hantro_v4l2.c                   58%     of 227
  hantro_vp8.c                    ---     of 17
  imx8m_vpu_hw.c                  ---     of 21
  rk3288_vpu_hw.c                 ---     of 39
  rk3399_vpu_hw.c                 ---     of 39
  rk3399_vpu_hw_jpeg_enc.c ---            of 69
  rk3399_vpu_hw_mpeg2_dec.c
  rk3399_vpu_hw_vp8_dec.c ---             of 109
▶ tee        ---       of 1315 ---        of 273
▶ thermal                         ---     of 3094
```

```
     }

     static void
     hantro_reset_raw_fmt(struct hantro_ctx *ctx)
     {
             const struct hantro_fmt *raw_vpu_fmt;
             struct v4l2_pix_format_mplane *raw_fmt, *encoded_fmt;

32           raw_vpu_fmt = hantro_get_default_fmt(ctx, false);

             if (hantro_is_encoder_ctx(ctx)) {
12                   ctx->vpu_src_fmt = raw_vpu_fmt;
                     raw_fmt = &ctx->src_fmt;
                     encoded_fmt = &ctx->dst_fmt;
             } else {
20                   ctx->vpu_dst_fmt = raw_vpu_fmt;
                     raw_fmt = &ctx->dst_fmt;
                     encoded_fmt = &ctx->src_fmt;
             }

32           hantro_reset_fmt(raw_fmt, raw_vpu_fmt);
             raw_fmt->width = encoded_fmt->width;
             raw_fmt->width = encoded_fmt->width;
             if (hantro_is_encoder_ctx(ctx))
12                   hantro_set_fmt_out(ctx, raw_fmt);
             else
20                   hantro_set_fmt_cap(ctx, raw_fmt);
32   }
```

FUJITSU

shaping tomorrow with you