HULK Robot助力Linux Kernel从开源到商用的转身

华为OS内核实验室

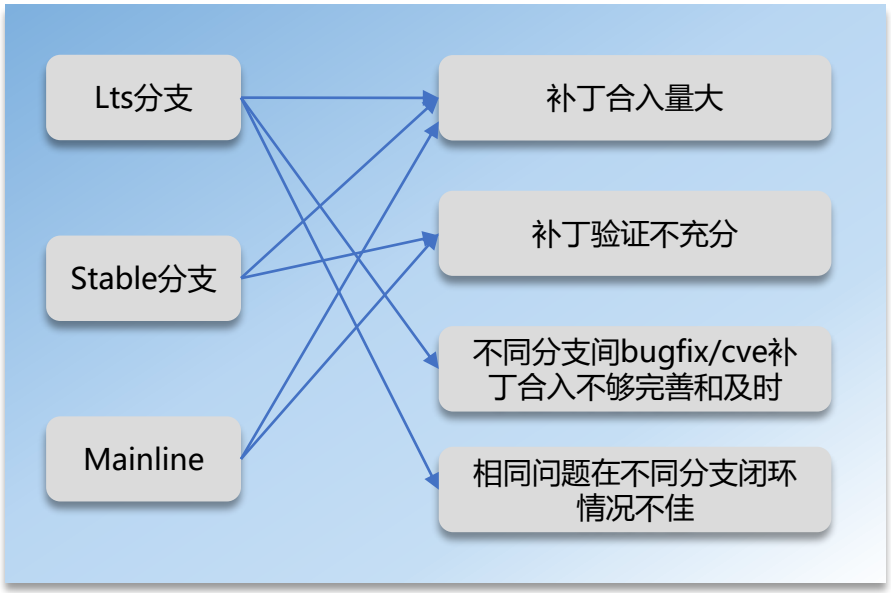
邹伟

目录

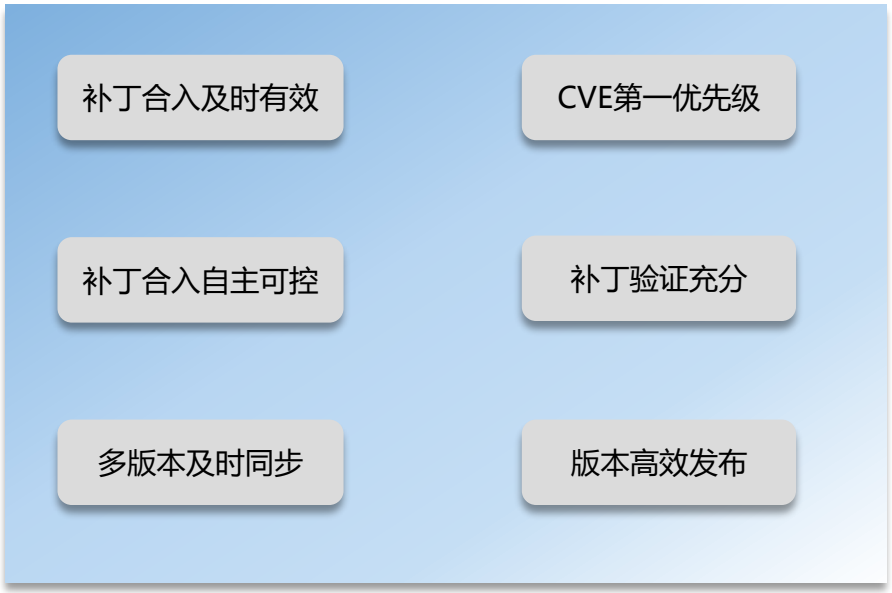
开源到商用的距离

OpenEuler

开源Kernel现状



商用Kernel诉求



➢ 社区Kernel版本从时效、质量、问题闭环完备性上无法满足商用要求

目录

商用版本的质量要求

OpenEuler

合入要求

规范：review机制、门禁..

正确：适配性、上下文检查...

流程可信

验证、合入过程需要可信

过程性数据、操作有据可查

流水线高效

验证方法高效，可重复性高

流水线作业，提高验证与发布及时性

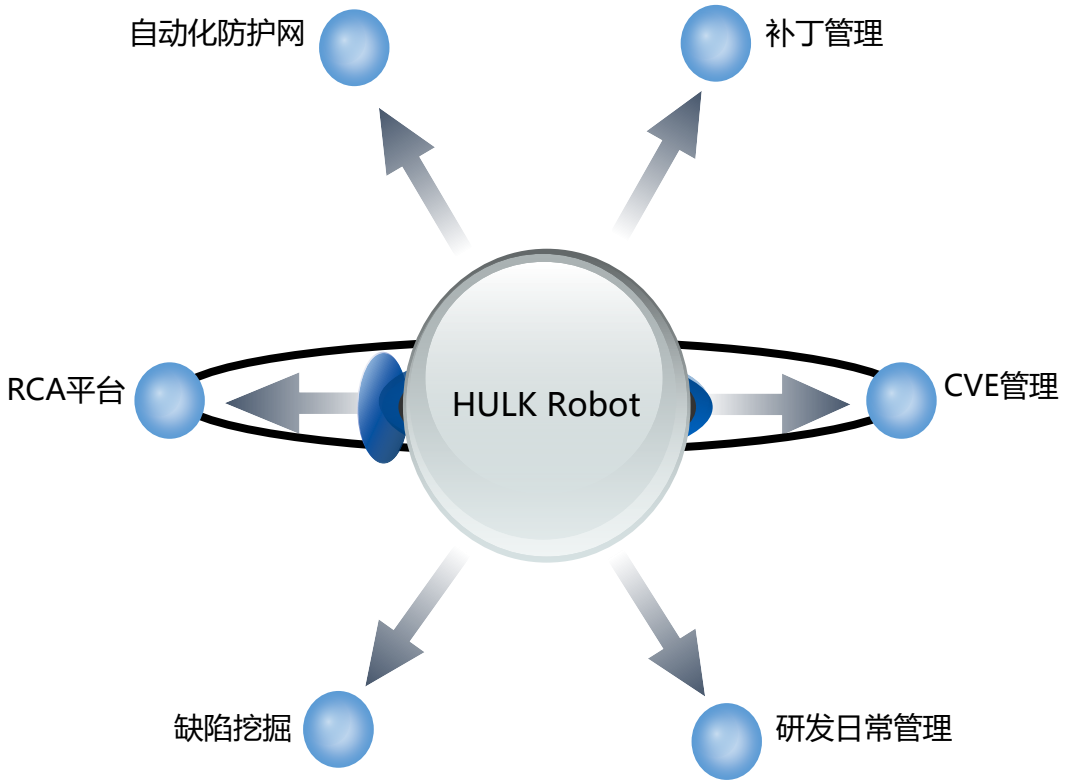
主动覆盖与挖掘

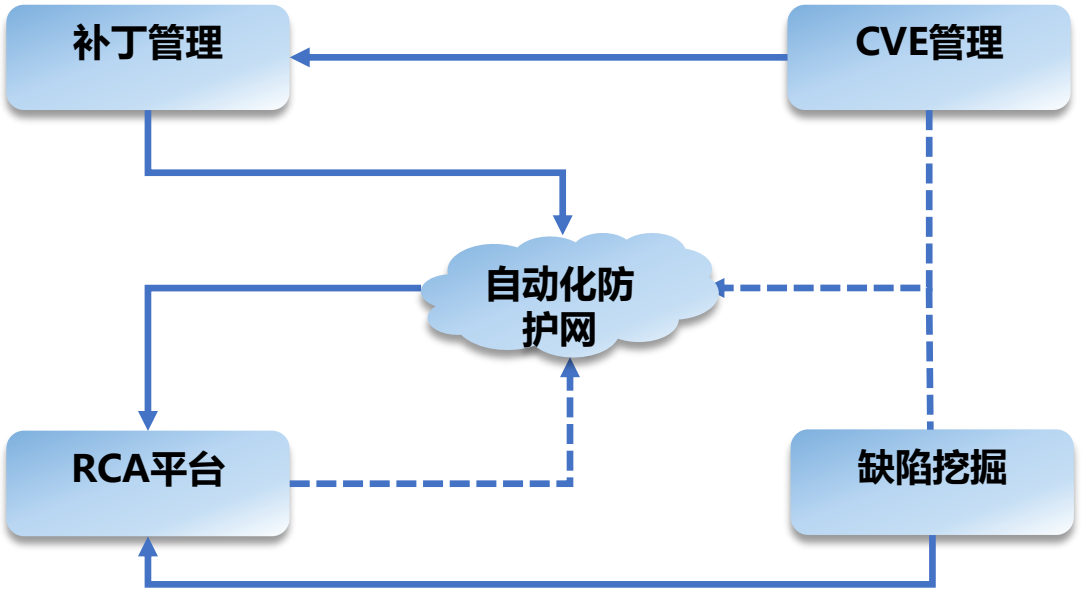
需要从功能、性能、可靠性、FUZZ等角度多手段进行验证

能够举一反三，防止同类问题重复出现

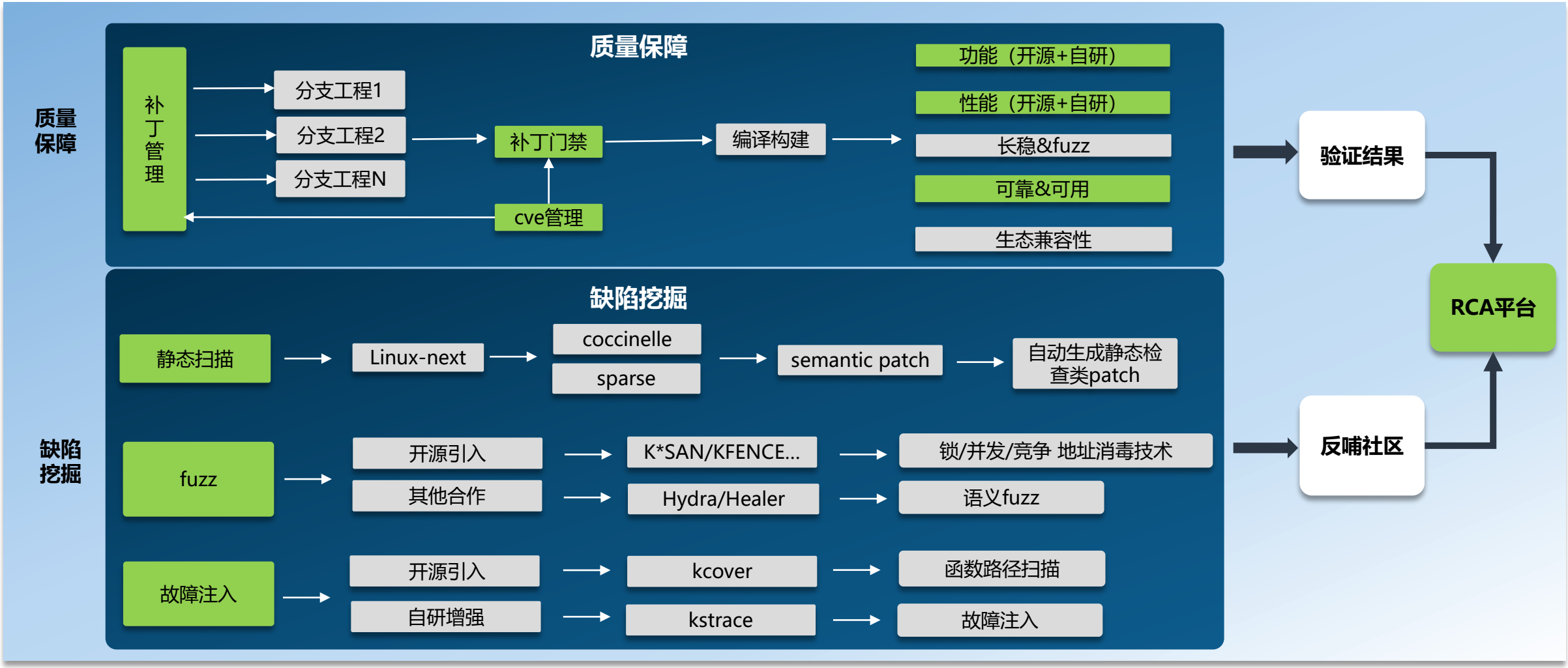
目录

HULK Robot的质量保障体系（总体）



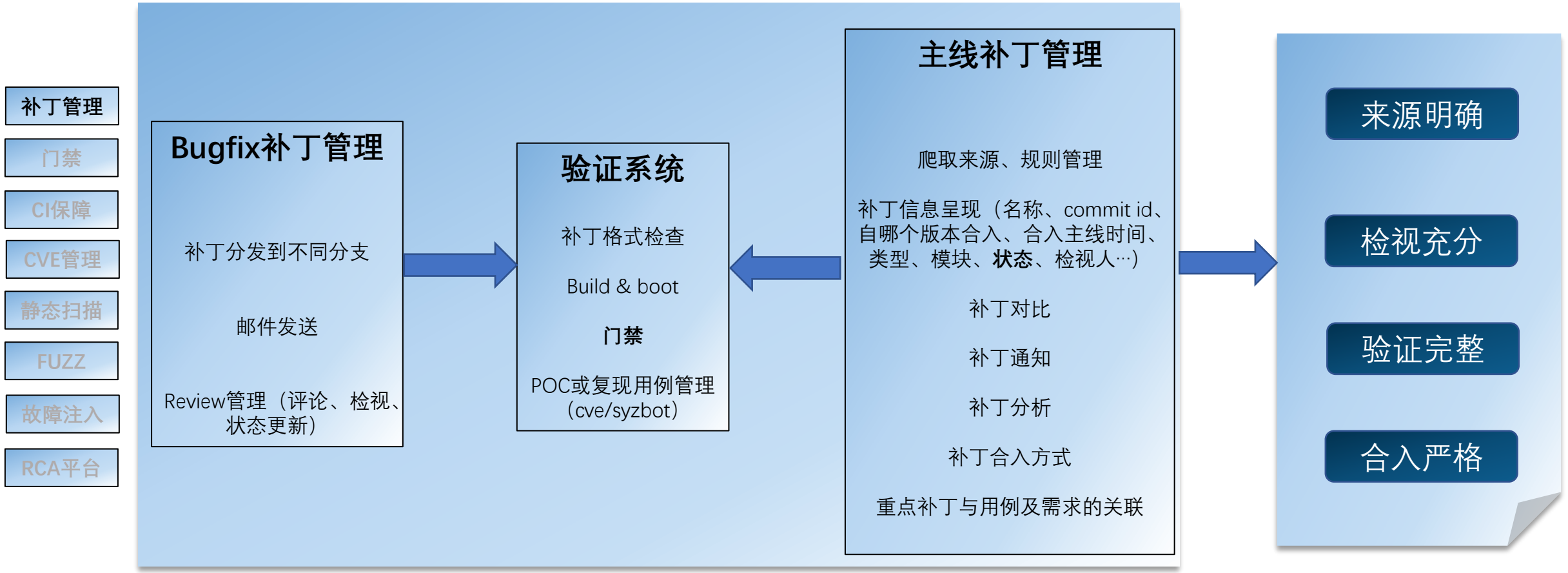


HULK Robot的质量保障体系



- 不同分支通过补丁门禁验证、CVE管理、编译构建后覆盖不同测试类型，保障版本质量
- 引入静态扫描、开源FUZZ工具、其他合作、通过覆盖率迁移、故障注入，挖掘并修复内核bug，反馈社区
- 在RCA平台里针对漏测问题进行根因分析，问题来源、根因类型、改进措施、关闭类型等

HULK Robot的质量保障体系（补丁管理）

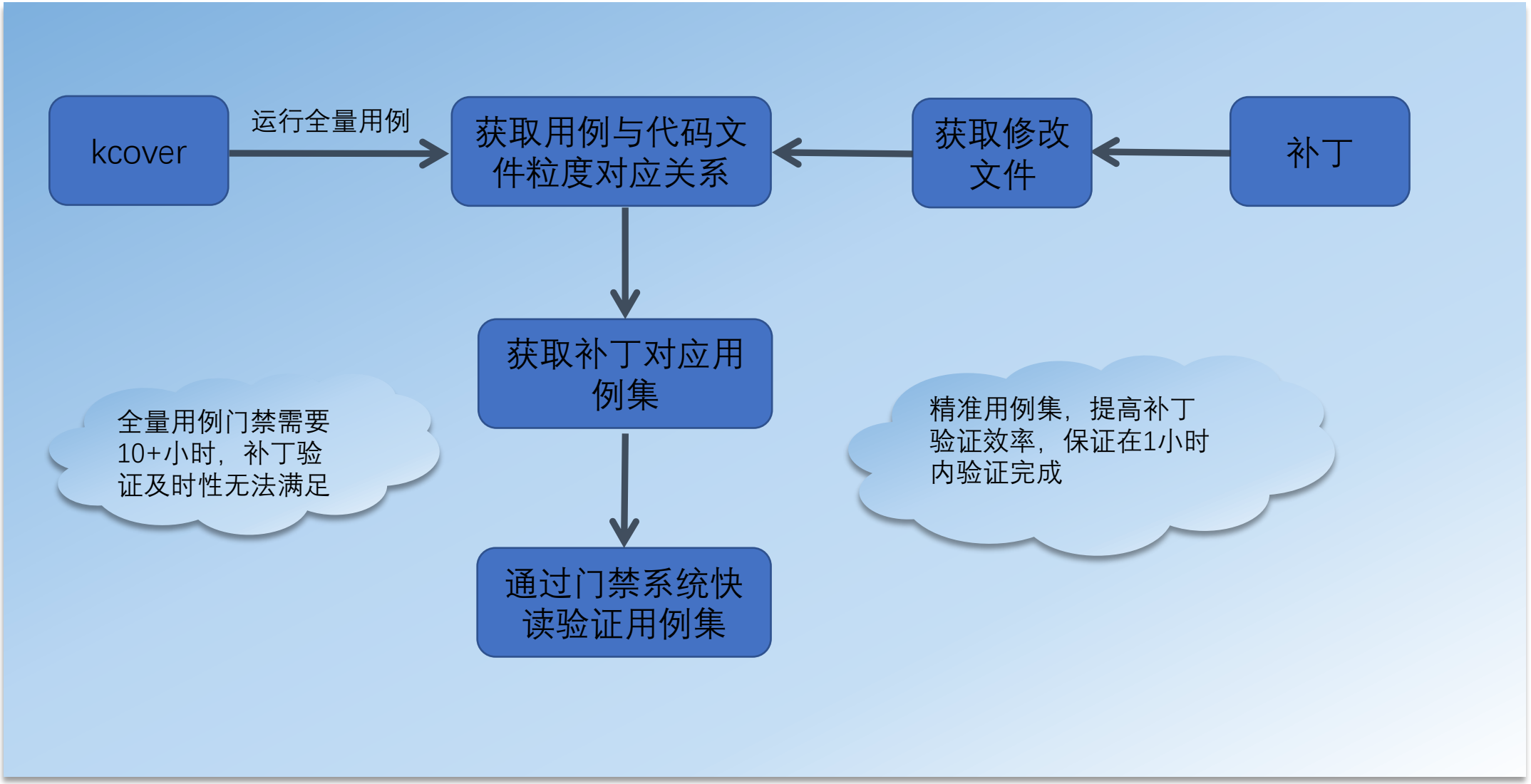


- LTS回合补丁已经够多，为什么还需要合入更多其他的补丁？
  交付场景多、需求多，需要合入更多的补丁来满足商用
- 帮助社区验证LTS 5.10/5.4/4.19/4.14，用例8900+
- 主动发现的bugfix问题，修复后回合到LTS

HULK Robot的质量保障体系（门禁）

OpenEuler

- 补丁管理
- 门禁
- CI保障
- CVE管理
- 静态扫描
- FUZZ
- 故障注入
- RCA平台



HULK Robot的质量保障体系（CI保障）

OpenEuler

补丁管理

门禁

CI保障

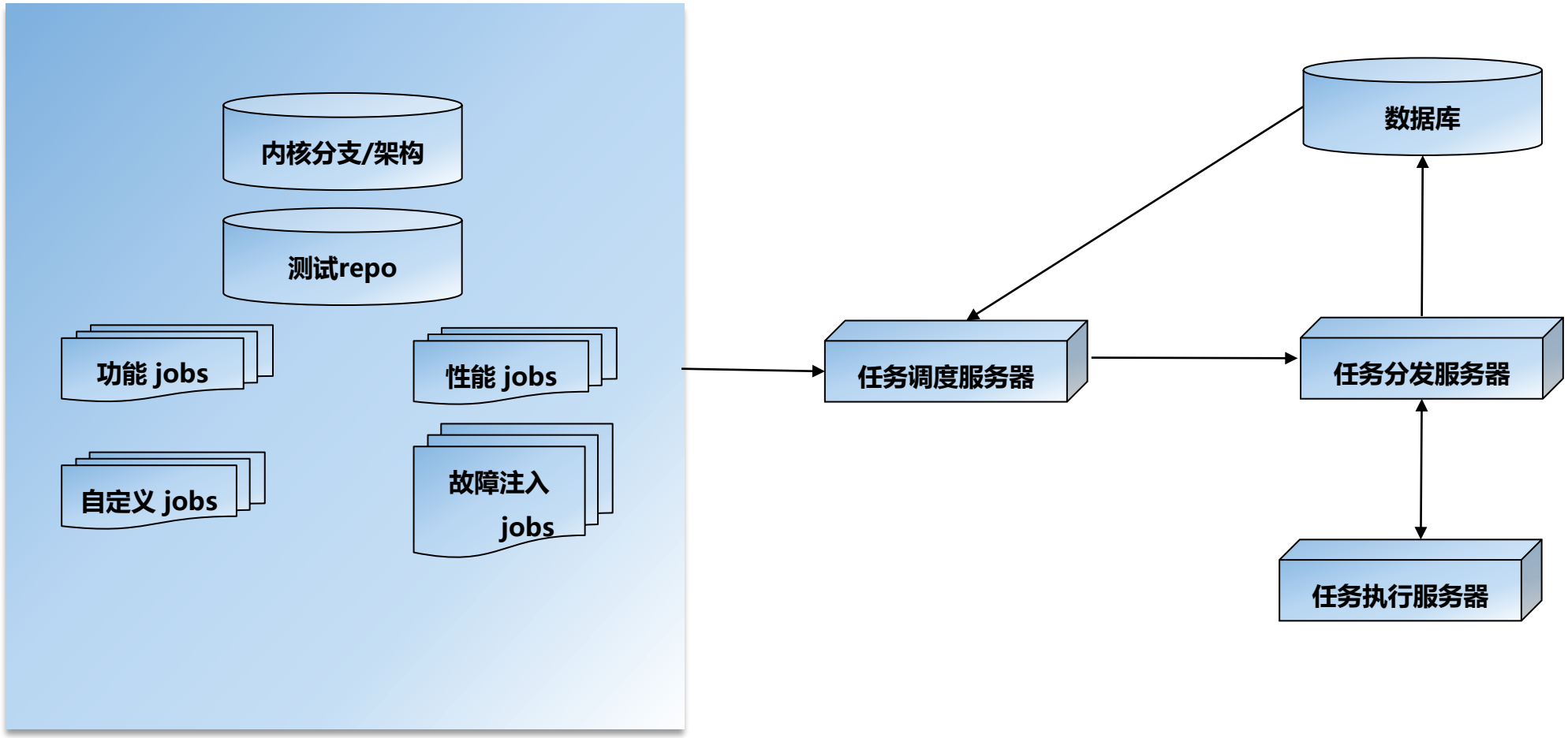
CVE管理

静态扫描

FUZZ

故障注入

RCA平台

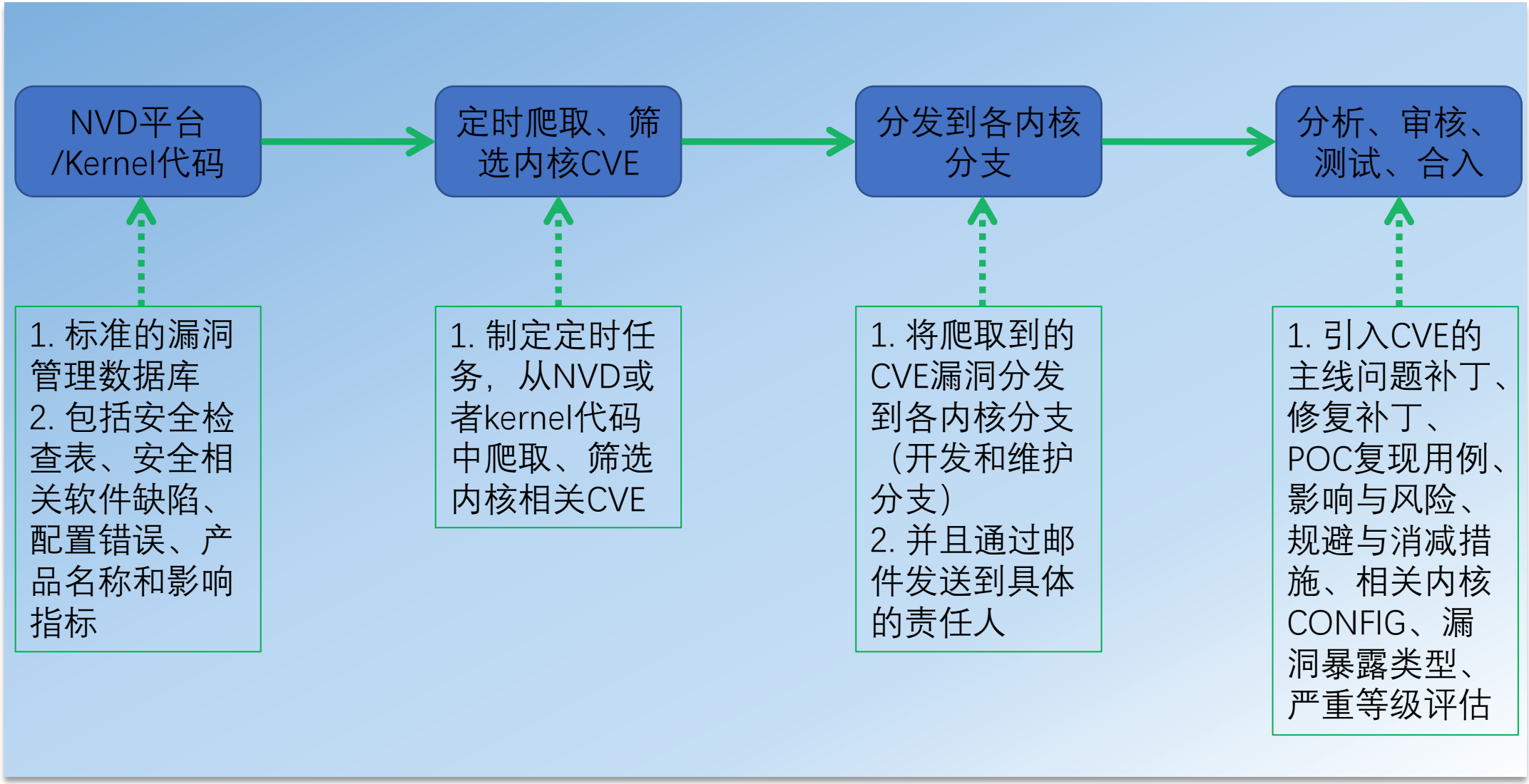


HULK Robot的质量保障体系（CVE管理）

OpenEuler

- 补丁管理
- 门禁
- CI保障
- **CVE管理**
- 静态扫描
- FUZZ
- 故障注入
- RCA平台

➢ CVE是第一优先级，LTS回合及时性无法满足商用交付效率，需要针对CVE专门进行管理



HULK Robot的质量保障体系（静态扫描）

OpenEuler

补丁管理
门禁
CI保障
CVE管理
静态扫描
FUZZ
故障注入
RCA平台

Linux-next.git
patch

sparse

通过 gcc 的扩展属性 __attribute__ 以及自定义的 __context__ 来对代码进行静态检查

make allmodconfig

ARCH: x86_64/arm64/ppc64/mips...

make C=2 drivers/***/***/***.o ARCH=***

coccinelle

程序的匹配和转换引擎、语言SMPL

negative-error-code-not-set.cocci
missing-pm-runtime-put.cocci
fwnode_handle_put.cocci

spatch -I include -sp_file ***.cocci ***.c

全量扫描（基线）+ 增量扫描

smatch
checkpatch
gcc/clang build warning
randconfigbuild
（x86_64/arm64/ppc..）
...

vchiq_arm.c:458:5: warning: symbol 'vchiq_platform_init' was not declared. Should it be static?

sem.c:575:9: warning: context imbalance in 'newary' - unexpected unlock

rockchip_i2s_tdm.c: warning: missing-pm-runtime-put.cocci

rzg2l_adc.c: warning: fwnode_handle_put.cocci

inode.c: warning: negative-error-code-not-set.cocci

自动生成patch
↓
补丁检查（格式、build&boot）
↓
sent to patchwork
↓
review检视
↓
send to kernel社区

HULK Robot的质量保障体系（静态扫描）（续）

OpenEuler

- 补丁管理
- 门禁
- CI保障
- CVE管理
- **静态扫描**
- FUZZ
- 故障注入
- RCA平台

规则范例

```
@@
struct file_operations fops@p = {
...,
.release = frelease,
...
};
@@
struct file_operations fops@p = {
...,
.open = fopen,
...,
+ .release = single_release,
};
@@
struct file_operations fops@p = {
...,
.open = fopen,
...,
- .release = frelease,
+ .release = single_release,
...
};
@@
struct file_operations fops@p = {
...,
.open = fopen,
...,
- .release = frelease,
...
};
```

Age	Commit message (Expand)
9 days	drm/nouveau/debugfs: fix file release memory leak
9 days	drm/nouveau/kms/nv50-: fix file release memory leak
2019-09-12	Merge tag 'gpio-v5.3-6' of git://git.kernel.org/pub/scm/linux/kernel/git/linu...
2019-09-09	crypto: cavium/zip - Add missing single_release()
2019-09-09	gpio: mockup: add missing single_release()
2019-09-06	rtlwifi: Fix file release memory leak
2019-09-06	rtw88: fix seq_file memory leak
2018-05-31	staging: rtlwifi: use single_open and single_release properly
2014-04-07	zram: factor out single stream compression
2013-11-27	cgroup: fix cgroup_subsys_state leak for seq_files
2013-07-31	Staging: rtl8192u/ieee80211: add missing single_release()
2013-07-31	Staging: rtl8192e: add missing single_release()
2013-05-05	Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/vir...
2013-04-29	fs/fscache/stats.c: fix memory leak
2012-05-21	drivers/net/stmmac: seq_file fix memory leak
2008-10-10	proc: fix return value of proc_reg_open() in "too late" case
2008-07-28	sh: fix seq_file memory leak

通过规则发现的问题

发现问题总结规则

Diffstat

-rw-r--r-- drivers/net/wireless/realtek/rtlwifi/debug.c 2

1 files changed, 1 insertions, 1 deletions

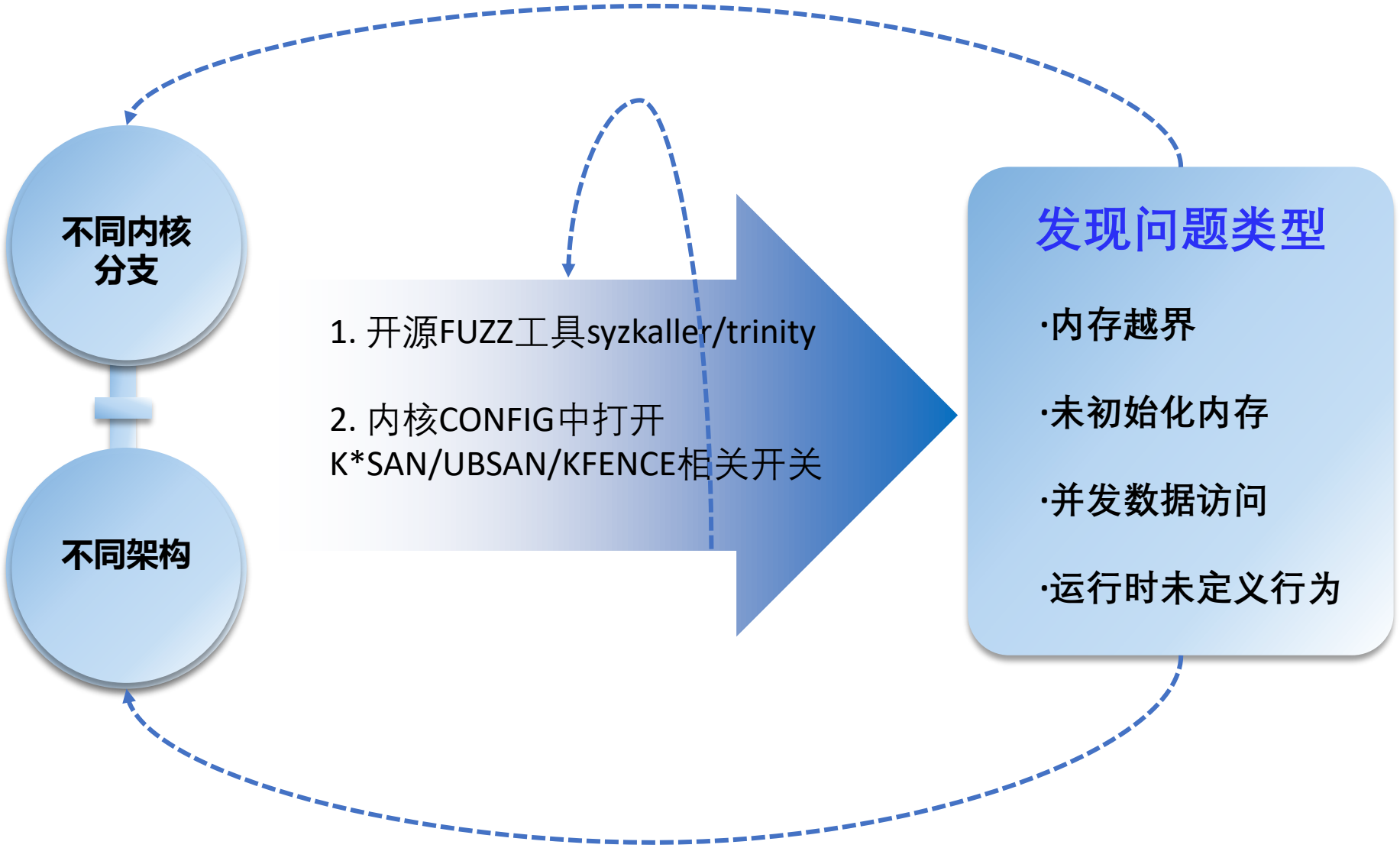
```
diff --git a/drivers/net/wireless/realtek/rtlwifi/debug.c b/drivers/net/wireless/realtek/rtlwifi/debug.c
index a051de16284df..55db71c766fe3 100644
--- a/drivers/net/wireless/realtek/rtlwifi/debug.c
+++ b/drivers/net/wireless/realtek/rtlwifi/debug.c
@@ -88,7 +88,7 @@ static const struct file_operations file_ops_common = {
        .open = dl_debug_open_common,
        .read = seq_read,
        .llseek = seq_lseek,
-       .release = seq_release,
+       .release = single_release,
 };

 static int rtl_debug_get_mac_page(struct seq_file *m, void *v)
```

补丁范例

HULK Robot的质量保障体系（FUZZ）

OpenEuler

- 补丁管理
- 门禁
- CI保障
- CVE管理
- 静态扫描
- FUZZ
- 故障注入
- RCA平台



HULK Robot的质量保障体系（故障注入）

基于ftrace特性在系统调用入口设置故障注入点
修改strace支持对故障类型、频率、时机等进行配置修改
打桩构造各类系统异常配置和执行上下文
组合存量场景用例开展故障注入测试
发现异常分支问题

示例：strace -o output.txt -e trace=mount -e inject=mount:when=1:fault=$i mount -o loop test.img /mnt umount /mnt

- 补丁管理
- 门禁
- CI保障
- CVE管理
- 静态扫描
- FUZZ
- 故障注入
- RCA平台

FAULT_INJECTION: forcing a failure. 指示在哪里注入了故障

```
[  717.580453] FAULT_INJECTION: forcing a failure.
[  717.580453] name failslab, interval 1, probability 0, space 0, times 0
[  717.593550] Call Trace:
[  717.594129]  dump_stack+0xce/0x12e
[  717.594903]  should_fail.cold+0x5/0xc
[  717.595735]  ? create_object+0x39/0xae0
[  717.596566]  should_failslab+0x5/0x10
[  717.597361]  kmem_cache_alloc+0x32f/0x460
[  717.598285]  create_object+0x39/0xae0
[  717.599097]  kmemleak_alloc_percpu+0xa0/0x100
[  717.600028]  pcpu_alloc+0x72d/0x10a0
[  717.600841]  __percpu_counter_init+0xdd/0x290
[  717.601769]  ext4_fill_super+0x6f64/0xb7a0
[  717.602747]  ? ext4_calculate_overhead+0x10a0/0x10a0
[  717.603811]  ? wait_for_completion+0x280/0x280
[  717.604794]  mount_bdev+0x2e8/0x3a0
[  717.605550]  ? ext4_calculate_overhead+0x10a0/0x10a0
[  717.606614]  ? ext4_free_in_core_inode+0x20/0x20
```

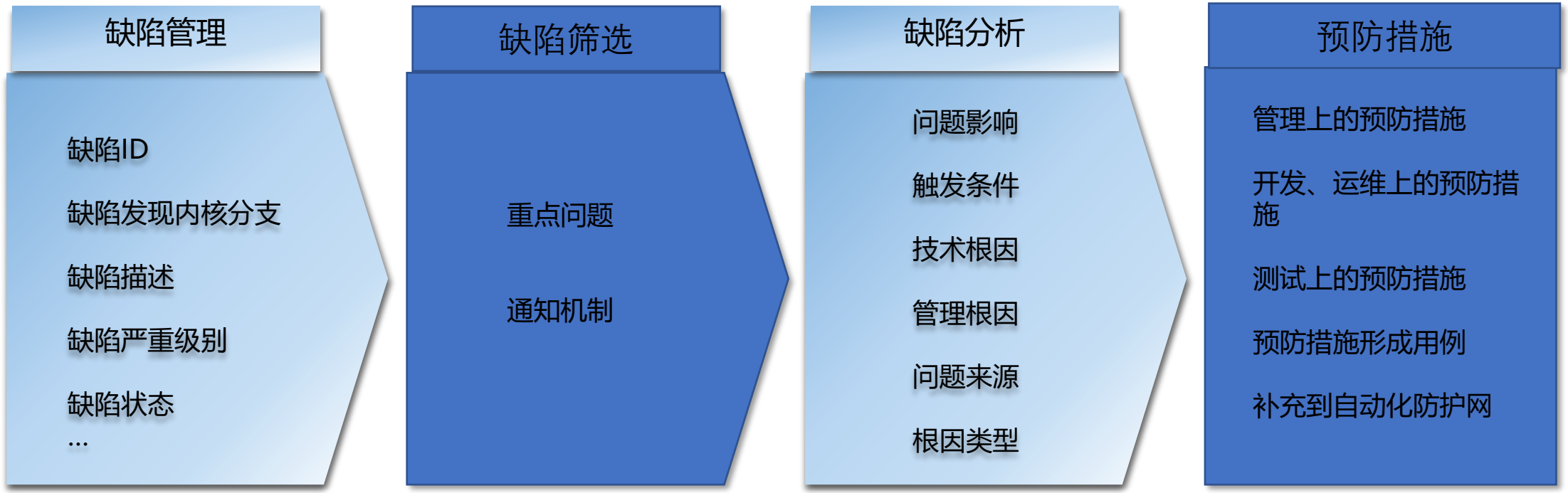
跑完结束，系统panic，指示空指针访问

```
[  723.755828] general protection fault, probably for non-canonical
address 0xdffffc000000001b: 0000 [#1] SMP KASAN PTI
[  723.759061] KASAN: null-ptr-deref in range
[0x00000000000000d8-0x00000000000000df]
[  723.766376] RIP: 0010:legacy_get_tree+0x12e/0x210
...
[  723.795386] Call Trace:
[  723.796570]  vfs_get_tree+0x8e/0x2d0
[  723.797631]  do_mount+0x1020/0x17f0
[  723.798463]  ? copy_mount_string+0x40/0x40
[  723.799646]  ? __might_fault+0x175/0x1b0
[  723.800629]  ? _copy_from_user+0xf1/0x150
[  723.801567]  ? memdup_user+0x62/0xb0
[  723.802314]  __x64_sys_mount+0x14b/0x1f0
[  723.803277]  do_syscall_64+0x56/0xa0
[  723.804014]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
```

HULK Robot的质量保障体系（RCA平台）

OpenEuler

- 补丁管理
- 门禁
- CI保障
- CVE管理
- 静态扫描
- FUZZ
- 故障注入
- RCA平台



- 缺陷根因充分剖析，举一反三，防止同类问题再次发生
- 减少因修改补丁引入的问题
- 减少因修复CVE引入的问题
- 减少已爬取补丁漏合的现象


OpenEuler