

Status:	CLOSED	Start date:	2023-06-07
Priority:	Urgent	Due date:	
Assignee:	CD TEST-方永红	% Done:	0%
Category:	重启	Estimated time:	0.00 hour
Target version:			
Need_Info:	--	Found Version:	0106
Resolution:	FIXED	Degrated:	--
Severity:	Critical	Verified Version:	
Reproducibility:	Rarely	Fixed Version:	
Test Type:	ST	Root cause:	
Description			
【环境信息】EE环境 【问题出现时间】：20230607 14点09分（第268次开机） 【前提条件】：车机上电（KL30ON，KL15 ON，已连接中控和功放，可正常输出声音） 副屏安装了十几个应用（27个支持的应用，比如QQ、QQ音乐HD、今日头条、企业微信、哔哩哔哩HD、应用宝、微信、快手、抖音、斗鱼、网易云音乐、网易新闻、腾讯视频、芒果TV、酷狗音乐、飞书等），副屏打开过以上应用，之后主要测试哔哩哔哩、爱奇艺、腾讯视频等页面时中控播放冥想动画 1.副屏点击桌面上的爱奇艺应用，显示黑屏没有进入爱奇艺主页面 2.副屏底侧向上滑动回到桌面， 3.再次点击爱奇艺应用 4.重复步骤2和步骤3多次 5.副屏底侧向上滑动 【实际结果】 副屏显示桌面，但无任何应用图标无状态栏显示，操作无效，之后自动重启，重启显示开机动画超过3分钟。 【期望结果】 副屏正常打开和使用爱奇艺，无异常重启现象。 【备注】： 车架号VIN：HRYTTESTVINMGM400，可从云端获取日志 副屏log地址：https://ofs.human-horizons.com/#/download/index/%2BMMVhkb4DTA%3D 视频地址：https://ofs.human-horizons.com/#/download/index/jGhO93UfRjw%3D 【影响版本】：副屏：01.00.0106.c104，中控：03.02.0088.c103 【硬件版本号】：/ 【是否能恢复】：			

ftp://cdiot@192.168.87.46/Pre_figure/Test_Log/118611

#2 - 2023-06-07 16:04 - CD FW 曹覃刚

- Status changed from New to ASSIGNED

- Assignee changed from CD FW 曹覃刚 to CD LC 陶洪普

■ 我的分析

在问题发生时，出现了大量的ANR

anr_2023-06-07-14-09-50-014

anr_2023-06-07-14-10-02-095

anr_2023-06-07-14-10-11-131

anr_2023-06-07-14-10-31-681

anr_2023-06-07-14-10-35-497

这些anr，里面的大部分线程都被下面这个线程block了，这个线程在处理网络相关的内容

"NetworkPolicy" prio=5 tid=75 Native | group="main" sCount=1 ucsCount=0 flags=1 obj=0x14143478 self=0xb400007c515987a0 | sysTid=1824 nice=0 cgrp=default sched=0/0 handle=0x7a08da9cb0 | state=S schedstat=(8413667679 2385392768 18791) utm=540 stm=301 core=3 HZ=100 | stack=0x7a08ca6000-0x7a08ca8000 stackSize=1039KB | held mutexes=

native: #00 pc 0000000000a1c88 /apex/com.android.runtime/lib64/bionic/libc.so (__ioctl+8)

native: #01 pc 000000000005b8d4 /apex/com.android.runtime/lib64/bionic/libc.so (ioctl+156)

native: #02 pc 0000000000053070 /system/lib64/libbinder.so (android::IPCThreadState::talkWithDriver(bool)+292)

native: #03 pc 00000000000542e4 /system/lib64/libbinder.so (android::IPCThreadState::waitForResponse(android::Parcel*, int*)+120)

native: #04 pc 0000000000053ff0 /system/lib64/libbinder.so (android::IPCThreadState::transact(int, unsigned int, android::Parcel const&, android::Parcel*, unsigned int)+224)

native: #05 pc 000000000004bafc /system/lib64/libbinder.so (android::BpBinder::transact(unsigned int, android::Parcel const&, android::Parcel*, unsigned int)+196)

native: #06 pc 0000000000167664 /system/lib64/libandroid_runtime.so (android_os_BinderProxy_transact(_JNIEnv*, _jobject*, int, _jobject*, _jobject*, int)+156)

at android.os.BinderProxy.transactNative(Native method)

at android.os.BinderProxy.transact(BinderProxy.java:571)

at android.net.INetd\$Stub\$Proxy.bandwidthRemoveInterfaceQuota(INetd.java:2997)

at com.android.server.NetworkManagementService.removeInterfaceQuota(NetworkManagementService.java:1083)

- locked <0x0b96493b> (a java.lang.Object)

at com.android.server.net.NetworkPolicyManagerService.removeInterfaceLimit(NetworkPolicyManagerService.java:5332)

at com.android.server.net.NetworkPolicyManagerService.access\$3400(NetworkPolicyManagerService.java:335)

at com.android.server.net.NetworkPolicyManagerService\$17.handleMessage(NetworkPolicyManagerService.java:5128)

at android.os.Handler.dispatchMessage(Handler.java:102)

at android.os.Looper.loopOnce(Looper.java:201)

at android.os.Looper.loop(Looper.java:288)

at android.os.HandlerThread.run(HandlerThread.java:67)

anr_2023-06-07-14-11-46-942

anr_2023-06-07-14-12-19-489

anr_2023-06-07-14-13-28-239

anr_2023-06-07-14-14-00-736

这些anr，system_server主线程在NetworkManagementService\$Dependencies.getNetd时sleep了

"main" prio=5 tid=1 Sleeping | group="main" sCount=1 ucsCount=0 flags=1 obj=0x715c36d8 self=0xb400007144c91be0 | sysTid=16038 nice=-2 cgrp=default sched=0/0 handle=0x7311dddf4f8 | state=S schedstat=(1101021355 359976293 3336) utm=77 stm=32 core=1 HZ=100 | stack=0x7fe7bfc000-0x7fe7bfe000 stackSize=8188KB | held mutexes=

at java.lang.Thread.sleep(Native method)

```
- sleeping on <0x050235a0> (a java.lang.Object)
at java.lang.Thread.sleep(Thread.java:451)
- locked <0x050235a0> (a java.lang.Object)
at java.lang.Thread.sleep(Thread.java:356)
at android.net.util.NetdService.get(NetdService.java:96)
at android.net.util.NetdService.get(NetdService.java:114)
at com.android.server.NetworkManagementService$Dependencies.getNetd(NetworkManagementService.java:127)
at com.android.server.NetworkManagementService.connectNativeNetdService(NetworkManagementService.java:432)
at com.android.server.NetworkManagementService.create(NetworkManagementService.java:265)
at com.android.server.NetworkManagementService.create(NetworkManagementService.java:271)
at com.android.server.SystemServer.startOtherServices(SystemServer.java:1813)
at com.android.server.SystemServer.run(SystemServer.java:885)
at com.android.server.SystemServer.main(SystemServer.java:616)
at java.lang.reflect.Method.invoke(Native method)
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:548)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:987)
```

SYSTEM PROCESS被杀了三次，由于主线程block在处理网络相关的内容

```
06-07 14:10:43.560 1521 1658 W Watchdog: * WATCHDOG KILLING SYSTEM PROCESS: Blocked in handler on ui thread (android.ui)
06-07 14:10:43.561 1521 1658 W Watchdog: android.ui annotated stack trace:
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.NetworkManagementService.setFirewallUidRule(NetworkManagementService.java:1579)
06-07 14:10:43.561 1521 1658 W Watchdog: - waiting to lock <0x0b96493b> (a java.lang.Object)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.net.NetworkPolicyManagerService.setUidFirewallRule(NetworkPolicyManagerService.java:5436)
06-07 14:10:43.561 1521 1658 W Watchdog: at
com.android.server.net.NetworkPolicyManagerService.updateRuleForAppIdleUL(NetworkPolicyManagerService.java:4280)
06-07 14:10:43.561 1521 1658 W Watchdog: at
com.android.server.net.NetworkPolicyManagerService.updateRulesForTempWhitelistChangeUL(NetworkPolicyManagerService.java:4435)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.net.NetworkPolicyManagerService.access$4400(NetworkPolicyManagerService.java:335)
06-07 14:10:43.561 1521 1658 W Watchdog: at
com.android.server.net.NetworkPolicyManagerService$NetworkPolicyManagerInternalImpl.onTempPowerSaveWhitelistChange(NetworkPolicyManagerService.java:576
8)
06-07 14:10:43.561 1521 1658 W Watchdog: - locked <0x07ef5858> (a java.lang.Object)
06-07 14:10:43.561 1521 1658 W Watchdog: at
com.android.server.DeviceIdleController.addPowerSaveTempWhitelistAppDirectInternal(DeviceIdleController.java:2931)
06-07 14:10:43.561 1521 1658 W Watchdog: at
com.android.server.DeviceIdleController$LocalService.addPowerSaveTempWhitelistAppDirect(DeviceIdleController.java:2071)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.am.ActivityManagerService.pushTempAllowlist(ActivityManagerService.java:14904)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.am.ActivityManagerService$UiHandler.handleMessage(ActivityManagerService.java:1646)
06-07 14:10:43.561 1521 1658 W Watchdog: at android.os.Handler.dispatchMessage(Handler.java:106)
06-07 14:10:43.561 1521 1658 W Watchdog: at android.os.Looper.loopOnce(Looper.java:201)
06-07 14:10:43.561 1521 1658 W Watchdog: at android.os.Looper.loop(Looper.java:288)
06-07 14:10:43.561 1521 1658 W Watchdog: at android.os.HandlerThread.run(HandlerThread.java:67)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.ServiceThread.run(ServiceThread.java:44)
06-07 14:10:43.561 1521 1658 W Watchdog: at com.android.server.UiThread.run(UiThread.java:45)
06-07 14:10:43.561 1521 1658 W Watchdog: * GOODBYE!

06-07 14:12:26.862 16038 16080 W Watchdog: * WATCHDOG KILLING SYSTEM PROCESS: Blocked in handler on main thread (main)
06-07 14:12:26.865 16038 16080 W Watchdog: main annotated stack trace:
06-07 14:12:26.865 16038 16080 W Watchdog: at java.lang.Thread.sleep(Native Method)
06-07 14:12:26.865 16038 16080 W Watchdog: at java.lang.Thread.sleep(Thread.java:451)
06-07 14:12:26.865 16038 16080 W Watchdog: - locked <0x050235a0> (a java.lang.Object)
06-07 14:12:26.865 16038 16080 W Watchdog: at java.lang.Thread.sleep(Thread.java:356)
06-07 14:12:26.865 16038 16080 W Watchdog: at android.net.util.NetdService.get(NetdService.java:96)
```

```
06-07 14:12:26.865 16038 16080 W Watchdog: at android.net.util.NetdService.get(NetdService.java:114)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.NetworkManagementService$Dependencies.getNetd(NetworkManagementService.java:127)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.NetworkManagementService.connectNativeNetdService(NetworkManagementService.java:432)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:265)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:271)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.SystemServer.startOtherServices(SystemServer.java:1813)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.SystemServer.run(SystemServer.java:885)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.server.SystemServer.main(SystemServer.java:616)
06-07 14:12:26.865 16038 16080 W Watchdog: at java.lang.reflect.Method.invoke(Native Method)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:548)
06-07 14:12:26.865 16038 16080 W Watchdog: at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:987)
06-07 14:12:26.865 16038 16080 W Watchdog: * GOODBYE!
```

```
06-07 14:14:08.111 16458 16479 W Watchdog: * WATCHDOG KILLING SYSTEM PROCESS: Blocked in handler on main thread (main)
06-07 14:14:08.113 16458 16479 W Watchdog: main annotated stack trace:
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Native Method)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Thread.java:451)
06-07 14:14:08.114 16458 16479 W Watchdog: - locked <0x08aa6b43> (a java.lang.Object)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Thread.java:356)
06-07 14:14:08.114 16458 16479 W Watchdog: at android.net.util.NetdService.get(NetdService.java:96)
06-07 14:14:08.114 16458 16479 W Watchdog: at android.net.util.NetdService.get(NetdService.java:114)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService$Dependencies.getNetd(NetworkManagementService.java:127)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.connectNativeNetdService(NetworkManagementService.java:432)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:265)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:271)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.startOtherServices(SystemServer.java:1813)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.run(SystemServer.java:885)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.main(SystemServer.java:616)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.reflect.Method.invoke(Native Method)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:548)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:987)
06-07 14:14:08.114 16458 16479 W Watchdog: * GOODBYE!
```

■ 下一步计划

上述3点内容，属于网络相关，需要LC的同学帮忙分析下

#3 - 2023-06-07 20:35 - 短距组_SSRDCNJ 王孟伟

//这里是导致黑屏并且重启的原因

```
06-07 14:14:08.111 16458 16479 W Watchdog: * WATCHDOG KILLING SYSTEM PROCESS: Blocked in handler on main thread (main)
06-07 14:14:08.113 16458 16479 W Watchdog: main annotated stack trace:
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Native Method)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Thread.java:451)
06-07 14:14:08.114 16458 16479 W Watchdog: - locked <0x08aa6b43> (a java.lang.Object)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.Thread.sleep(Thread.java:356)
06-07 14:14:08.114 16458 16479 W Watchdog: at android.net.util.NetdService.get(NetdService.java:96)
06-07 14:14:08.114 16458 16479 W Watchdog: at android.net.util.NetdService.get(NetdService.java:114)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService$Dependencies.getNetd(NetworkManagementService.java:127)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.connectNativeNetdService(NetworkManagementService.java:432)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:265)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.NetworkManagementService.create(NetworkManagementService.java:271)
```

06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.startOtherServices(SystemServer.java:1813)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.run(SystemServer.java:885)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.server.SystemServer.main(SystemServer.java:616)
06-07 14:14:08.114 16458 16479 W Watchdog: at java.lang.reflect.Method.invoke(Native Method)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.internal.os.RuntimeInit\$MethodAndArgsCaller.run(RuntimeInit.java:548)
06-07 14:14:08.114 16458 16479 W Watchdog: at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:987)
06-07 14:14:08.114 16458 16479 W Watchdog: * GOODBYE!
06-07 14:14:08.114 16458 16479 I Process : Sending signal. PID: 16458 SIG: 9
06-07 14:14:08.209 658 673 I mpu_uart: [MSG-P:RECV]:No message received in 1000 ms
06-07 14:14:08.227 516 516 I lowmemorykiller: Imkd data connection dropped
06-07 14:14:08.227 516 516 I lowmemorykiller: closing Imkd data connection
06-07 14:14:08.231 30570 31799 W iorapd : Reconnect to package manager service: 1 times
06-07 14:14:08.234 30570 31799 I ServiceManager: Waiting for service 'package_native' on '/dev/binder'...
//重启
06-07 14:14:08.235 16418 16418 E Zygote : Zygote failed to write to system_server FD: Connection refused
06-07 14:14:08.235 16418 16418 I Zygote : Process 16458 exited due to signal 9 (Killed)
06-07 14:14:08.235 16418 16418 E Zygote : Exit zygote because system server (pid 16458) has terminated

//黑屏的重启原因是因为应用程序无法访问所需的网络管理功能，即无法获取到netd

06-07 14:14:07.149 16458 16458 W NetdService: WARNING: returning null INetd instance.

建议测试人员尝试重启设备并清除应用程序缓存复测，

开机后开机动画持续三分钟，log中没找到相关block日志，请system团队或fw团队帮忙检查开机动画持续三分钟的原因

#4 - 2023-06-07 21:11 - CDTs-TEST 周婷

- Assignee changed from CD LC 陶洪普 to CD FW 曹覃刚

请帮忙继续分析

#5 - 2023-06-07 22:42 - CD LC 陶洪普

- Assignee changed from CD FW 曹覃刚 to CD LC 陶洪普

#反馈的设备重启

从kernel日志中\log\tlog_iiov0201017500006162025411a2212290000000911_000269_0607142452\bk\FSEM_000269_0838_0607141433.tar\kernel\未看到整机的重启，有一次动画启动

06-07 14:10:43.833 0 0 I [14988.889578] init: starting service 'bootanim'...

06-07 14:10:43.858 0 0 I [14988.906507] init: Control message: Processed ctl.start for 'bootanim' from pid: 900 (/system/bin/surfaceflinger)

1. 上面反馈的三次因获取不到Net instance, 而导致 watchdog kill system service, 是反馈的重启后一直在开机动画中现象。

之所以获取不到netd (netd启动过程中未发现明显错误日志), 系统的低内存影响了netd启动运行 (迟迟未完成启动)

详细请参照 <https://thundersoft.feishu.cn/docx/LxaSdAAsGolYfTxooimc80hAnoh> 分析

下一步，请system解决低内存问题。

#6 - 2023-06-07 22:55 - CD LC 陶洪普

- Assignee changed from CD LC 陶洪普 to CD FW 曹覃刚

Hi 覃刚，胡兵

请确认上面分析和问题发生时间点：

- 1. #06-07 14:10:43 系统有一次重启动画，时间是连续的。未看到整机重启，请帮分析触发原因
- 2. 在06-07 14:10:44.587 之后，*有三次netd的重启，先是低内存影响了netd启动，进而影响了获取netd instance*。请review问题发生的先后顺序。

详细请确认 <https://thundersoft.feishu.cn/docx/LxaSdAAsGolYfTxxoimc80hAnoh>

谢谢！

#7 - 2023-06-08 11:47 - CD SYSTEM-夏旭

- 1. #06-07 14:10:43 系统有一次重启动画，时间是连续的。未看到整机重启，请帮分析触发原因

--> 因为system_server watchdog 导致android 上层 java世界全部重启，包括zygote,当时的ANR文件，查看

tlog_0607142452\FSEM_000269_0838_0607141433\fatal\anr\anr_2023-06-07-14-10-35-497

06-07 14:10:43.545 1521 15926 I DropBoxManagerService: add tag=system_server_watchdog isTagEnabled=true flags=0x6

06-07 14:10:43.560 1521 1658 W Watchdog: *** WATCHDOG KILLING SYSTEM PROCESS: Blocked in handler on ui thread (android.ui)

- 2. 在06-07 14:10:44.587 之后，*有三次netd的重启，先是低内存影响了netd启动，进而影响了获取netd instance*。请review问题发生的先后顺序。
开机脚本中，指定了，当zygot重启时会，重启netd 这个很正常

```
service zygote /system/bin/app_process64 -Xzygote /system/bin --zygote --start-system-server
class main
priority -20
user root
group root readproc reserved_disk
socket zygote stream 660 root system
socket usap_pool_primary stream 660 root system
onrestart exec_background - system system -- /system/bin/vdc volume abort_fuse
onrestart write /sys/power/state on
onrestart restart audioserver
onrestart restart cameraserver
onrestart restart media
onrestart restart netd ---》会导致netd重启
onrestart restart wificond
writepid /dev/cpuset/foreground/tasks
critical window=${zygote.critical_window.minute:-off} target=zygote-fatal
```

#8 - 2023-06-08 15:46 - CD FW 曹覃刚

- Assignee changed from CD FW 曹覃刚 to CD SYSTEM-夏旭

Hi 洪普 夏旭

结合上述你们的分析

当前应该需要看第一次 netd启动被block 导致上层重启的原因

CD LC 陶洪普

之所以获取不到netd (netd启动过程中未发现明显错误日志)，系统的低内存影响了netd启动运行（迟迟未完成启动）

下一步，请system解决低内存问题。

需要system的同学帮助分析，低内存的原因

CD SYSTEM-夏旭

开机脚本中，指定了，当zygot重启时会，重启netd 这个很正常

zygot重启时会重启netd，这个是正常的，异常的地方在于启动netd一直启动不成功，LC的同事分析是系统低内存导致，因此需要检查系统低内存的原因

下一步，应该是需要system的同学检查系统低内存的原因
注:飞书拉了一个此票的对话，有问题可以在飞书中及时沟通，谢谢

#9 - 2023-06-08 20:39 - CD SYSTEM-胡兵
- Assignee changed from CD SYSTEM-夏旭 to CD LC 陶洪普

Hi 洪普

如群里沟通，目前问题分成两类，均是网络相关：

第一次重启： NetworkManagementService ,- waiting to lock <0x0b96493b> (a java.lang.Object)
需要调查谁在持锁，block主进程
后面两次重启： android.net.util.NetdService.get触发watchdog
从日志中看，netd启动慢的直接原因为，netd listing chain时出现异常报错，需要继续调查

./out/000269_merged_logcat_file(3).log:620881:06-07 14:13:32.397 16423 16423 E Netd : Error listing chain OUTPUT in table filter

./out/000269_merged_logcat_file(3).log:621639:06-07 14:13:42.422 16423 16423 E Netd : Error listing chain OUTPUT in table filter

./out/000269_merged_logcat_file(3).log:622416:06-07 14:13:52.458 16423 16423 E Netd : Error listing chain POSTROUTING in table mangle

./out/000269_merged_logcat_file(3).log:623398:06-07 14:14:02.521 16423 16423 E Netd : Error listing chain POSTROUTING in table mangle

#10 - 2023-06-09 17:42 - CD TPM-王祥林
- Status changed from ASSIGNED to NEED_INFO

转到jira票上更新

#11 - 2023-06-17 18:04 - CD TEST-方永红

6.17

【复现版本】

/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230609/202306091536-4284/FlatBuild_HH_MCE_FSE.M.D.user.01.00.C104%28X104%29.202306091536.zip

【复现情况】

- 1、打开爱奇艺、查看页面显示
- 2、上划关闭爱奇艺
- 3、打开爱奇艺、查看页面显示
- 4、重复上述步骤500次（先手动100次，然后使用脚本重复400次）

最终结果：未复现

#12 - 2023-06-20 22:05 - IoT scm

Gerrit Merge Information :

ID	Project	Branch	Uploader
208143	general/platform/system/netd	Pre_figure_turbox-c2130c-la1.1-qssi12-dev	taohp0107@thundersoft.com
Network: Kill the process if it does not exit with the timeout periodAdd thread to check if the process is exited by kill(SIGTERM) inIptablesProcess->stop. If no, call kill(SIGKILL) to kill it to avoidblocking main process.TC-RID: 1200-0400701IssueID: TS-R-BUG-118611Change-Id: I4f42065ce12a85196d2946e6bf0d7d3315d19d4d			

#13 - 2023-06-25 21:40 - IoT scm

Gerrit Merge Information :

ID	Project	Branch	Uploader
209670	general/platform/system/netd	Pre_figure_turbox-c2130c-la1.1-qssi12-release_sorp	wangxl0510@thundersoft.com
Network: Kill the process if it does not exit with the timeout periodAdd thread to check if the process is exited by kill(SIGTERM) inIptablesProcess->stop. If no, call kill(SIGKILL) to kill it to avoidblocking main process.TC-RID: 1200-0400701IssueID: TS-R-BUG-118611Change-Id: I4f42065ce12a85196d2946e6bf0d7d3315d19d4d(cherry picked from commit c1b446333c44e7d30b0b4875767585584bf40a06)			

#14 - 2023-06-28 19:50 - CD LC 陶洪普

- Assignee changed from CD LC 陶洪普 to CD TEST-方永红

已提交，转JIRA更新

<http://jira.human-horizons.com/browse/IOVDEV-33202>

#15 - 2023-07-10 16:45 - CD TEST-方永红

- Status changed from NEED_INFO to RESOLVED

- Resolution changed from -- to FIXED

#16 - 2023-07-10 16:46 - CD TEST-方永红

- Status changed from RESOLVED to VERIFIED

7.10

客户验证通过，同步关闭

#17 - 2023-07-10 16:46 - CD TEST-方永红
- Status changed from VERIFIED to CLOSED