Android 10

STS 测试操作指南

1.0

2020.02.26

文档履历

版本号	日期	制/修订人	内容描述
1.0	2020.02.26		Android 10 STS 测试操作指南

目录

1. 测试环境搭建

1.1 获取 STS 测试包

STS 全称 Security Test Suite。是 Google 提供的针对安全的测试套件。每月会有一次更新，用于检测系统的重要安全补丁是否已经添加。STS 测试相关的环境，配置，测试命令和结果获取均与 CTS 非常相似。但 STS 测试工具是不开源的，由 GMS 送测渠道提供。

1.2 Ubuntu 主机测试环境搭建

对测试主机的具体要求如下：

1. 安装 Ubuntu16.04 LTS 64bit 系统

2. 添加 adb 工具，配置系统使主机能够通过 adb 连接平板。

3. 安装 JDK1.8。

4. 安装 aapt 工具。

5. 安装 STS 测试工具。将 1.1 节中获取的 zip 压缩文件解压到测试主机中，解压后得到名字为 android-sts 的文件夹，注意不要修改此文件夹及其子目录和文件的名字。

6. 主机需要连接外网。

注意：需要设置环境变量，使系统能够找到 adb，java 版本和 aapt 工具。

1.3 网络环境

STS 测试过程中会连接国外的网络如：youtube。使用国内的网络无法访问这些网站。需要使用能正常访问这些网站的网络环境进行测试。

测试过程会测试 ipv6 网络，所以 wifi 需要支持 ipv6。

2. 测试前平板配置

平板固件烧录完成后需要进行相关配置才能测试 STS。进行 STS 测试的设备都必须是安全机器，并已经烧写 google attestation key。

主要的配置如下：

1. 恢复出厂设置。这一项在有必要的情况下进行。刚烧好固件运行起来的可跳过此步骤。如果烧录好固件后被用作其它用途再进行 STS 测试，则需要先恢复出厂设置或重烧固件。

2. 在设置中选择语言为 English(United States)。

3.Settings > Display > Brightness 设置为最小。

4.Settings > Display > Sleep 设置最长休眠时间。

5. 选项 Settings > Security > Screen Lock 为 none，确保设备上未设置锁定图案或密码。

6. 选项 Setting> Wi-fi, 连接支持 ipv6 和能连接国外网络的网络。

7. 勾选 USB 调试。Settings > Developer options > USB debugging。(注意，在 4.2 之后的系统中 Developer options 默认是不显示的，需要进入 Settings > About tablet，然后迅速连续敲击 Build number 七次，返回上一级菜单查找开发者选项)。

8. 勾选 Settings > Developer options > Stay Awake，保持屏幕常亮。

9. 去掉勾选 Settings > Developer options > Verify apps over USB。

10. 连接能够上国外网络的 Wifi AP。

14. 将平板通过 USB 连接到测试主机。在 USB debugging 弹框中勾选 Always allow from this computer，点击 OK。

注意：

必须使用 userdebug 固件测试。如果测试报告需要提交 Google 或者 3PL，STS 使用的 userdebug 固件 fingerprint 需要和测试 CTS 时使用的 user 固件的 fingerprint 基本一致。目前 Android 10 的代码，只需要在编译前，输入 export BUILD_NUMBER=$(date +%Y%m%d%H%M)，然后依次编译 user，userdebug 固件，用 adb 命令 adb shell getprop |grep fingerprint 检查两者的 fingerprint 是否一致。

3. 启动 STS 测试

3.1 STS 完整测试

启动 sts 测试，需要在终端进入 1.2 步骤所解压的 android-sts 文件夹下的 tools 目录下，执行命令./sts-tradefed，会启动测试平台。STS 目前测试需要的时间不到 3 个小时，可以使用一台机器完成测试。

3.1.1 完整测试指令

测试命令：run sts-engbuild

参数：

-s ：平板序列号可通过"l d"（list device 的首字母缩写）命令查看

--logcat-on-failure ： 抓取 fail 项 log

--shard-count x : 使用 x 台机器并行测试，STS 建议用一台测试即可。

--abi(-a) arm64-v8a 或者 --abi(-a) armeabi-v7a ： 指定测试 64/32 系统

比如：

run sts-engbuild --shard-count 1 -s <平板序列号 1> --logcat-on-failure

3.2 STS 补测

3.2.1 Retry 测试

除谷歌允许的 FAIL 外, 如果 STS 测试 fail 项超过允许的 FAIL 项，要进行补测。

测试命令：

run retry --retry <session_id> -s <serial>

session_id: 可通过"l r" 命令查看

比如通过如下命令启动补测：

run retry -r session_id --shard-count 1 -s 平板序列号 1

3.2.2 创建补测测试计划以及启动测试计划

Add: 可以通过 help add 命令查看帮助

a/add s/subplan: create a subplan from a previous session

Options:

--session : The session used to create a subplan.

--name/-n : The name of the new subplan.

--result-type : Which results to include in the subplan. One of passed, failed, not_executed.Repeatable.

例：a s --session 2 --name subplan_name --result-type failed --result-type not_executed

通过如下命令启动补测：

run sts-engbuild --subplan subplan_name（subplan_name : 上述中创建的名字）

3.3 针对性测试

可以针对某个测试包，测试类或者具体测试用例进行测试。如下图所示。

run <plan> --module/-m <module> --test/-t <test_name>: run a specific test from the module. Test name can be <package>.<class>, <package>.<class>#<method> or <native_name>.

例：

1．测试整个 module

run sts-engbuild -m CtsSecurityTestCases

2. 测试整个 class

run sts-engbuild -m CtsSecurityTestCases -t android.security.cts.AmbiguousBundlesTest

3. 测试一项 test

run sts-engbuild -m CtsSecurityTestCases -t android.security.cts.AmbiguousBundlesTest#test_android_CVE_2017

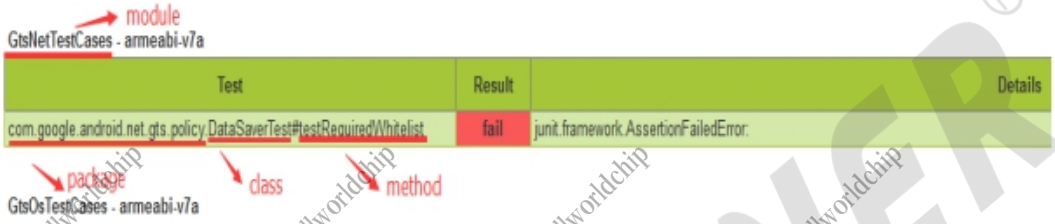

图 1: 测试 module, class, testcase 分布

3.4 跳过某些有问题的测试项

比如测试的时候 CtsSecurityTestCases 项目出现了问题，导致机器卡死或者测试中断，导致无法进行其他的项目的测试，这个时候可以选择跳过测试该项目，在执行的命令加上：

--exclude-filter CtsSecurityTestCases 或者 --exclude-filter "CtsSecurityTestCases

android.security.cts.AmbiguousBundlesTest#test_android_CVE_2017_0806"，注意，跳过某个用例时，模块与用例名要用空格隔开，且要有双引号包住。

其他与此类推，跳过多项时，重复输入参数。

3.5 循环测试

如需多次执行测试，无需等待正在执行的测试完成，直接输入多次测试命令即可，这些命令会被缓存起来，被依次调用。

4. 测试结果分析与调试

4.1 从 STS 报告中获取出错信息

测试后的结果保存在 android-sts\results 目录下，可以通过浏览器打开 test_result_failures_suite.html 文件显示出测试的结果，在错误项的的 Details 栏中给出了基本的错误信息。

测试过程抓取的 log 保存在 android-sts\logs 目录下。

4.2 查找相关 log 信息

测试时加上 --logcat-on-failure 参数，sts 工具会自动将 fail 的测试 log 保存在 android-sts\logs 下，打开 log 文件，搜索 TestRunner。测试过程的 log 以 ``TestRunner: started:…." 开始，以 ``TestRunner: finished: ….." 结束。

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application.tates nor implies warranty of any kind, including fitness for any particular application.