

数据库备份

实验概述

在Android设备中，正常情况下应用程序文件保存在/data/data/<app 'package name>/目录中。如果应用程序使用数据库存取数据，就会在包名目录下有一个databases文件夹。本实验将所有应用数据库文件备份到pc中。

实验目的

- 1、了解busybox的功能
- 2、熟练adb命令
- 3、熟练find、tar等命令

实验原理

安卓虽然是构建在Linux Kernel上的手机操作系统，但是精简了很多linux工具，很多常用的linux指令不能使用。busybox相当于一个打包的工具箱，打包了很多的常用的linux可执行文件和其依赖。安装了busybox你就可以在安卓下下载一个模拟终端然后在里面运行一些之前不能运行的指令。BusyBox 是一个集成了一百多个最常用linux命令和工具的软件。其中包含了一些简单的工具，例如ls、cat和echo等等，还包含了一些更大、更复杂的工具，例grep、find、mount、tar以及telnet。有些人将 BusyBox 称为Linux 工具里的瑞士军刀。在安卓系统下，运用了数据库保存数据的应用，一般数据库文件会保存在“data/data/包名”这个目录当中，但是在android系统中阉割了许多linux命令导致了无法对文件进行复制、压缩等操作，因此需要用busybox工具箱里面的命令对文件进行操作。

find命令

语法格式：find path -option [-print] [-exec -ok command] {} \;

path: find命令所查找的目录路径。例如用.来表示当前目录，用/来表示系统根目录。

-print: find命令将匹配的文件输出到标准输出。

-exec: find命令对匹配的文件执行该参数所给出的shell命令。相应命令的形式为'command' {} \;，注意{}和\;之间的空格。

-ok: 和-exec的作用相同，只不过以一种更为安全的模式来执行该参数所给出的shell命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。

-name #查找名为filename的文件

-perm #按执行权限来查找

-user #按文件属主来查找

-group #按组来查找

-mtime -n +n #按文件更改时间来查找文件，-n指n天以内，+n指n天以前

-atime -n +n #按文件访问时间来找GIN: 0px">

-ctime -n +n #按文件创建时间来查找文件, -n指n天以内, +n指n天以前

-nogroup #查无有效属组的文件, 即文件的属组在/etc/groups中不存在

-nouser #查无有效属主的文件, 即文件的属主在/etc/passwd中不存

-ctime -n +n #按文件创建时间来查找文件, -n指n天以内, +n指n天以前

-nogroup #查无有效属组的文件, 即文件的属组在/etc/groups中不存在

-nouser #查无有效属主的文件, 即文件的属主在/etc/passwd中不存

-newer f1 !f2 #查更改时间比f1新但比f2旧的文件

-type b/d/c/p/l/f #查是块设备、目录、字符设备、管道、符号链接、普通文件

-size n[c] #查长度为n块[或n字节]的文件

-depth #使查找在进入子目录前先行查找完本目录

-fstype #查更改时间比f1新但比f2旧的文件

-type b/d/c/p/l/f #查是块设备、目录、字符设备、管道、符号链接、普通文件

-size n[c] #查长度为n块[或n字节]的文件

-depth #使查找在进入子目录前先行查找完本目录

-prune #忽略某个目录

tar命令

语法格式: tar [主选项+辅选项] 文件或者目录

主选项c 创建新的档案文件。如果用户想备份一个目录或是一些文件, 就要选择这个选项。相当于打包。

x 从档案文件中释放文件。相当于拆包。

t 列出档案文件的内容, 查看已经备份了哪些文件。

辅助选项:

-z : 是否同时具有 gzip 的属性,亦即是否需要用 gzip 压缩或解压 .一般格式为xx.tar.gz或xx. tgz

-j : 是否同时具有 bzip2 的属性,亦即是否需要用 bzip2 压缩或解压.一般格式为xx.tar.bz2

-v : 压缩的过程中显示文件

-f : 使用档名

-p : 使用原文件的原来属性 (属性不会依据使用者而变)

--exclude FILE: 在压缩的过程中, 不要将 FILE 打包!

Sqlitebrowser

sqlitebrowser是一款免费、开源、免安装的可视化数据库浏览管理软件。它能够用最简单、直观的方式创建、编辑、处理SQL数据库。可以创建需要的数据库并分别为其创建图表, 百分百自定义创建。当然, 还可以进行内容编辑以及SQL查询。创建好数据库之后, 可以将其输出为CSV或SQL文件。

实验环境

虚拟机: kali linux

工具: adb、sqlitebrowser、busybox

模拟器: android 4.0

实验步骤

1、“打开终端”>“cd android-sdk-linux/tools/”>“./android”来启动android sdk
如图 1

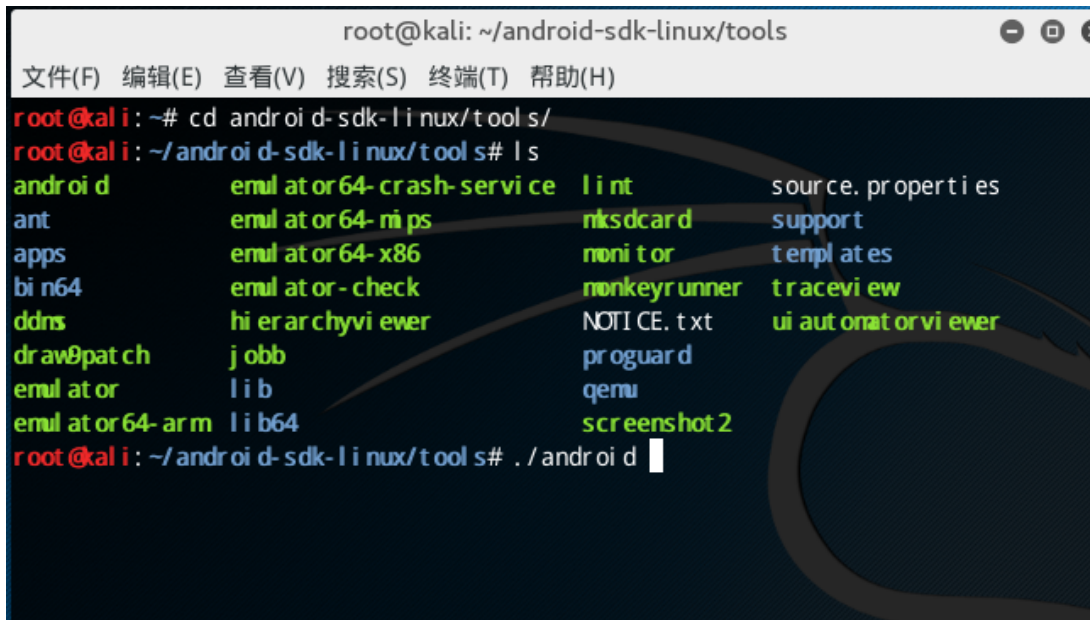


图 1 开启android sdk

2、“单击tools”>“选择Manage AVD”打开虚拟机控制台, 如图 2

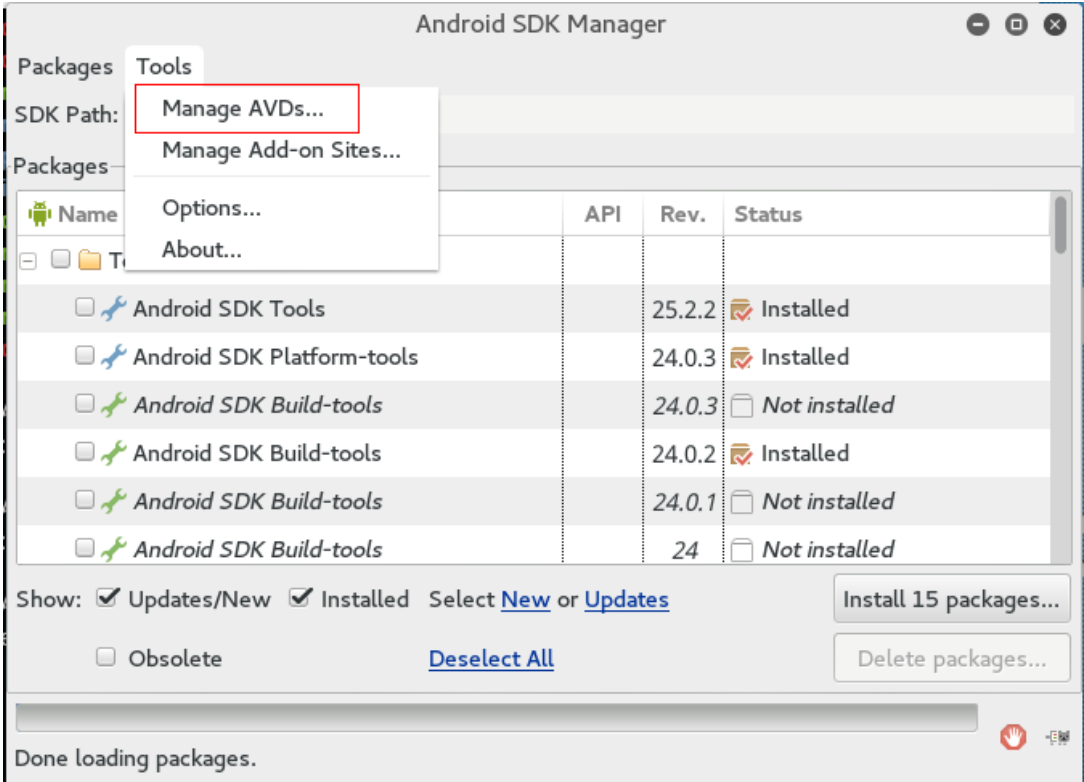


图 2开启模拟器控制台

3、选择创建好的Android虚拟机单击start来开启虚拟机，如图 3

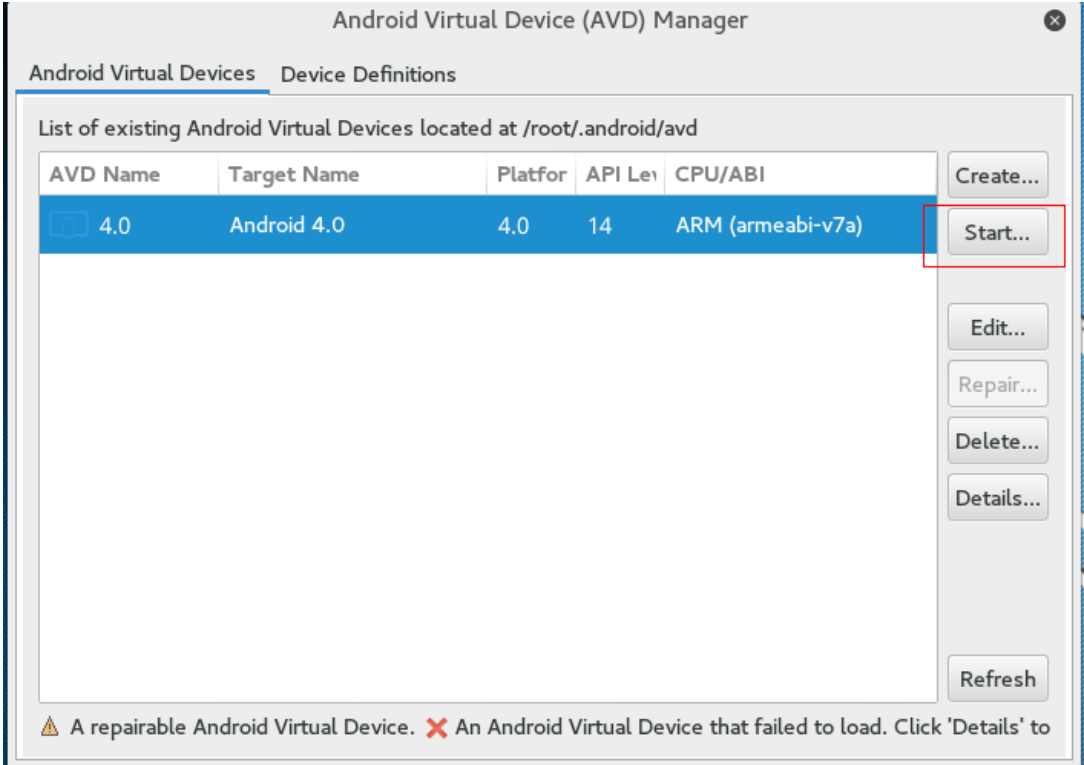


图 3开启模拟器

4、此处可设置屏幕的尺寸，使用默认值，单击launch，如图 4

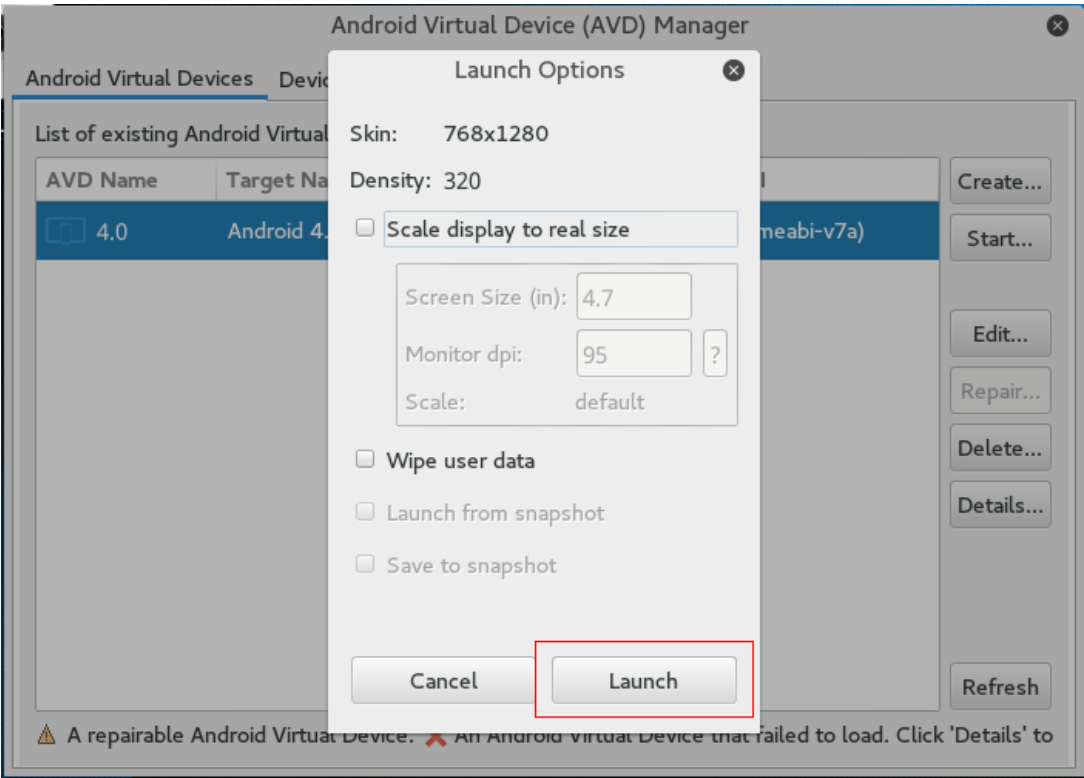


图 4开启模拟器

5、成功开启android虚拟机，桌面如图 5

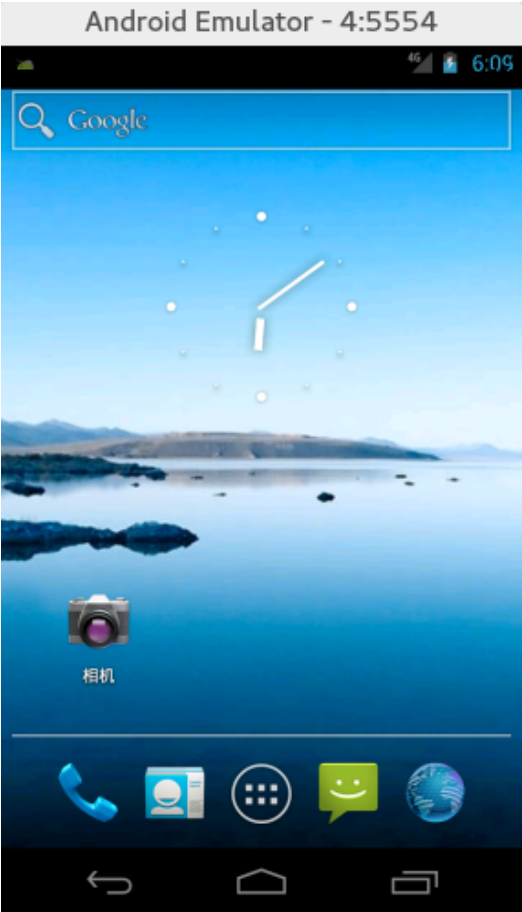


图 5模拟器界面

6、修改安卓system文件可写入。

操作：“打开终端”>输入“adb shell”>“mount -o remount,rw -t yaffs2 /dev/block/mtdblock3 /system”如图 6

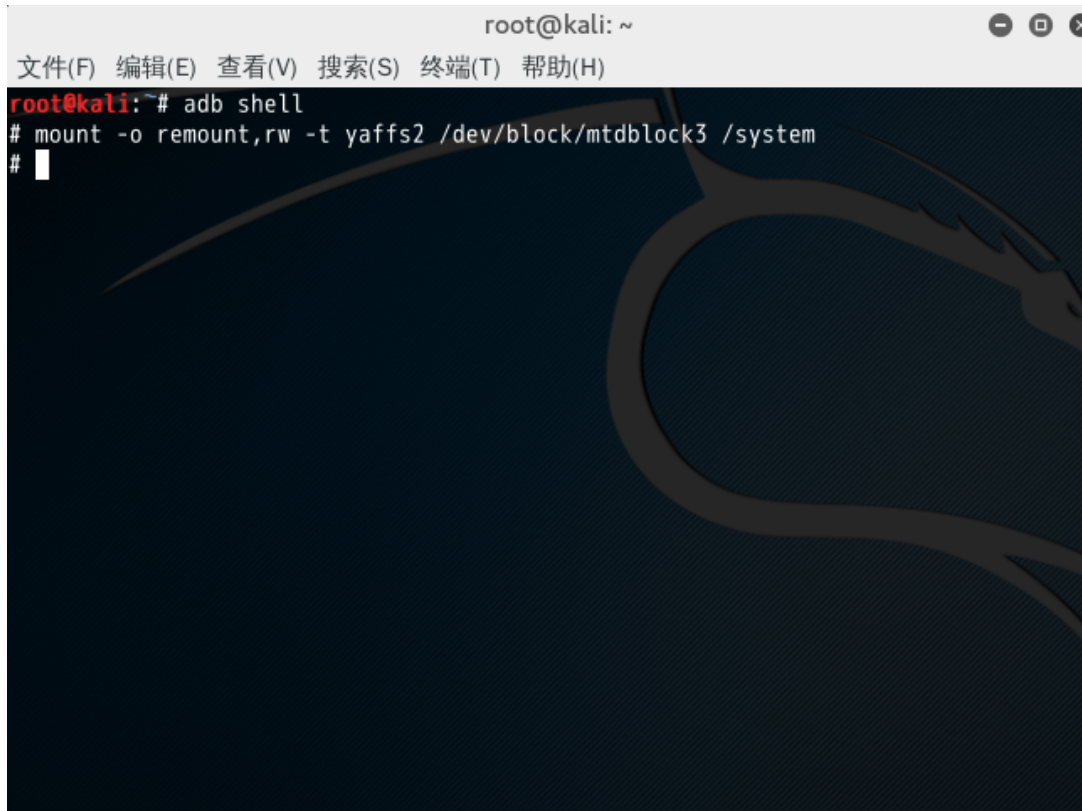


图 6重新挂载

7、把本地的busybox上传到xbin目录中。

操作：“新建终端”>“输入命令:adb push /root/tools/busybox /system/xbin”如图 7

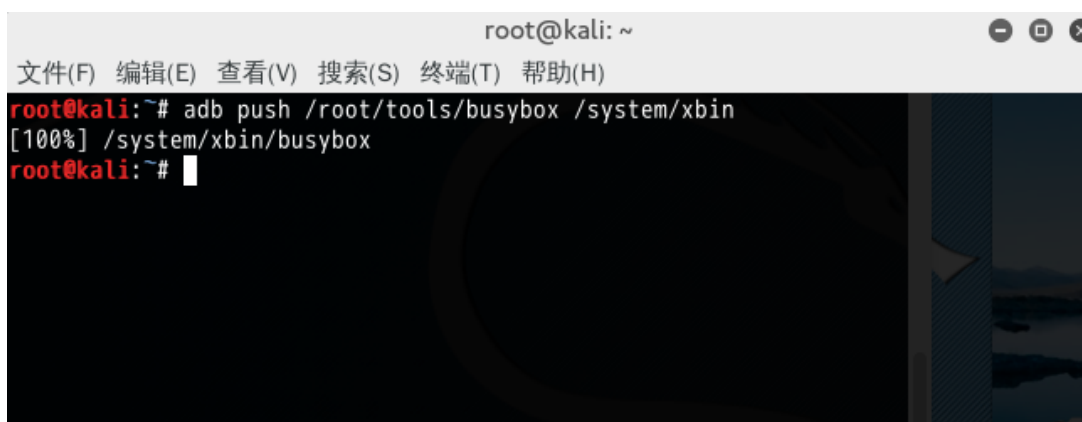
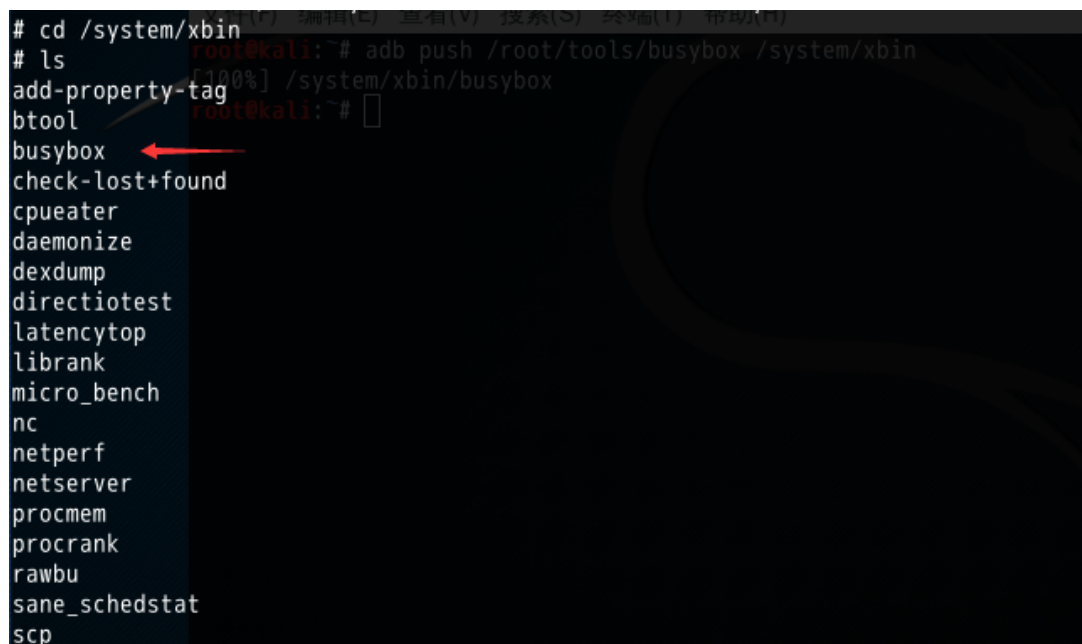


图 7上传busybox

8、查看是否上传成功。

操作：“cd /system/xbin”>“ls”。如图 8

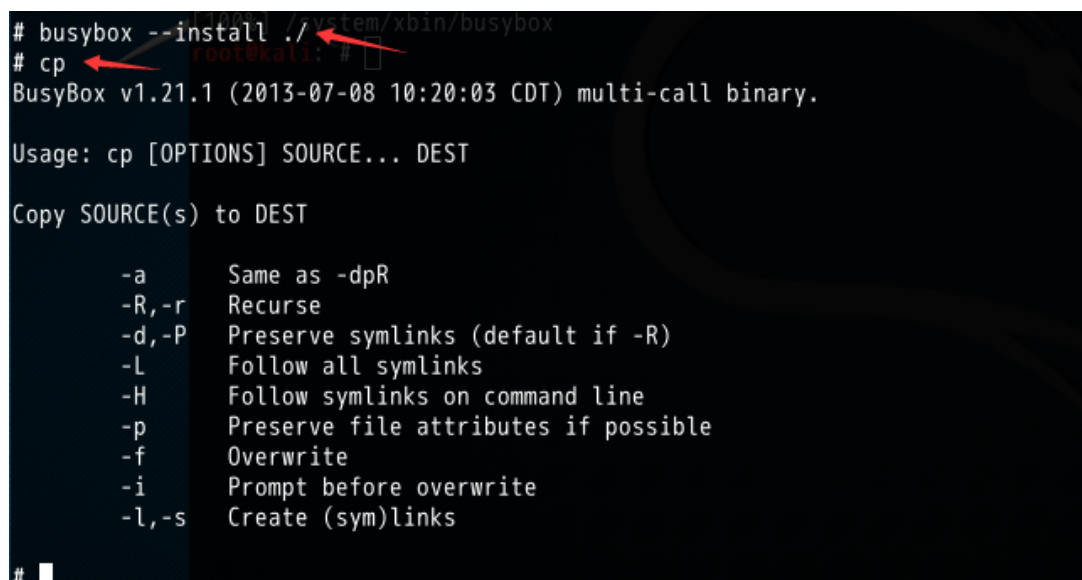


```
# cd /system/xbin
# ls
add-property-tag
btool
busybox
check-lost+found
cpueater
daemonize
dexdump
directiotest
latencytop
librank
micro_bench
nc
netperf
netserver
procmem
procrank
rawbu
sane_schedstat
scp
```

图 8上传成功

9、对busybox进行安装，并且查看是否安装成，输入cp命令显示如图 9结果表示安装成功

操作：“busybox --install ./”>“cp”



```
# busybox --install ./
# cp
BusyBox v1.21.1 (2013-07-08 10:20:03 CDT) multi-call binary.

Usage: cp [OPTIONS] SOURCE... DEST

Copy SOURCE(s) to DEST

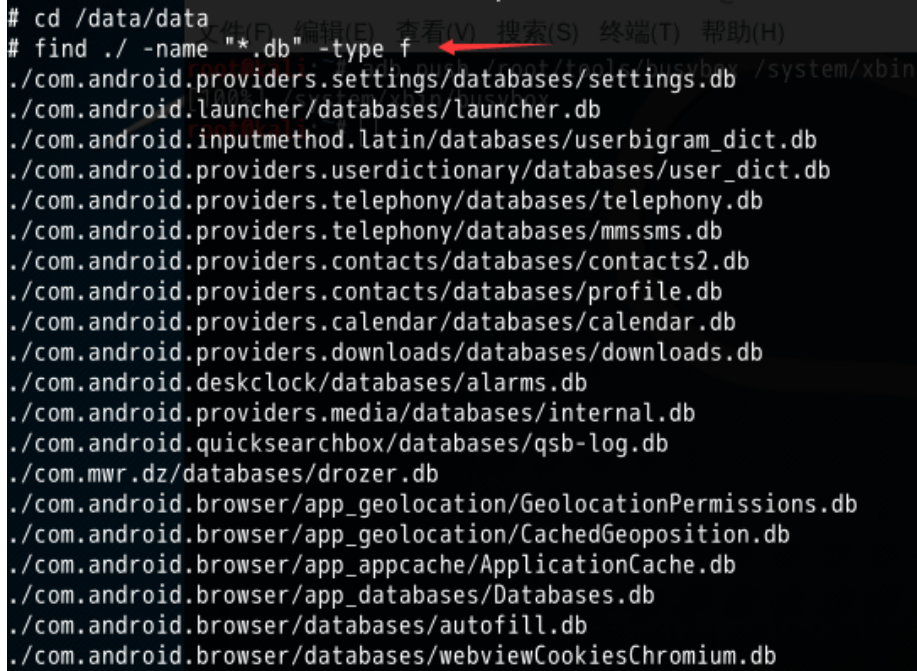
    -a      Same as -dpR
    -R,-r   Recurse
    -d,-P   Preserve symlinks (default if -R)
    -L      Follow all symlinks
    -H      Follow symlinks on command line
    -p      Preserve file attributes if possible
    -f      Overwrite
    -i      Prompt before overwrite
    -l,-s   Create (sym)links

#
```

图 9安装成功

10、进入到data目录，查看数据库文件。

操作：“cd /data/data”>“find ./ -name “*.db” -type f”如图 10

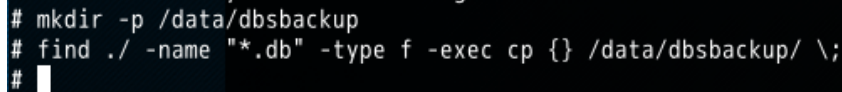


```
# cd /data/data
# find ./ -name "*.db" -type f
./com.android.providers.settings/databases/settings.db
./com.android.launcher/databases/launcher.db
./com.android.inputmethod.latin/databases/userbigram_dict.db
./com.android.providers.userdictionary/databases/user_dict.db
./com.android.providers.telephony/databases/telephony.db
./com.android.providers.telephony/databases/mmssms.db
./com.android.providers.contacts/databases/contacts2.db
./com.android.providers.contacts/databases/profile.db
./com.android.providers.calendar/databases/calendar.db
./com.android.providers.downloads/databases/downloads.db
./com.android.deskclock/databases/alarms.db
./com.android.providers.media/databases/internal.db
./com.android.quicksearchbox/databases/qslog.db
./com.mwr.dz/databases/drozer.db
./com.android.browser/app_geolocation/GeolocationPermissions.db
./com.android.browser/app_geolocation/CachedGeoposition.db
./com.android.browser/app_appcache/ApplicationCache.db
./com.android.browser/app_databases/Databases.db
./com.android.browser/databases/autofill.db
./com.android.browser/databases/webviewCookiesChromium.db
```

图 10匹配寻找db数据库文件

11、创建备份目录并且把数据库文件复制到备份目录中。

操作：“mkdir -p /data/dbsbackup”>“find ./ -name “*.db” -type f -exec cp {} /data/dbsbackup/ \;”如图 11



```
# mkdir -p /data/dbsbackup
# find ./ -name "*.db" -type f -exec cp {} /data/dbsbackup/ \;
#
```

图 11复制到备份目录

12、进入到备份目录，查看是否备份成功。

操作：“cd /data/dbsbackup/”>“ls”如图 12


```

# cd /data/dbsbackup/
# ls
ApplicationCache.db
CachedGeoposition.db
Databases.db
EmailProvider.db
EmailProviderBackup.db
EmailProviderBody.db
GeolocationPermissions.db
WebpageIcons.db
alarms.db
autofill.db
browser2.db
calendar.db
contacts2.db
downloads.db
drozer.db
grants.db
internal.db
launcher.db
mmssms.db
profile.db

```

图 12 备份成功

13、对备份目录进行压缩。

操作：“cd ..”>“tar -zcvf backups.tar dbsbackup”如图 13

```

# tar -zcvf backups.tar dbsbackup
dbsbackup/
dbsbackup/grants.db
dbsbackup/EmailProvider.db
dbsbackup/EmailProviderBackup.db
dbsbackup/EmailProviderBody.db
dbsbackup/WebpageIcons.db
dbsbackup/browser2.db
dbsbackup/webview.db
dbsbackup/webviewCookiesChromiumPrivate.db
dbsbackup/webviewCookiesChromium.db
dbsbackup/autofill.db
dbsbackup/Databases.db
dbsbackup/ApplicationCache.db
dbsbackup/CachedGeoposition.db
dbsbackup/GeolocationPermissions.db
dbsbackup/drozer.db
dbsbackup/qsb-log.db
dbsbackup/internal.db

```

图 13 压缩备份目录

14、查看是否压缩成功。如图 14



图 14压缩成功

15、把备份文件下载到本机并且增加其执行权限。

操作：“adb pull /data/backups.tar ./”>“ls”>“chmod 770 backups.tar”>“ls”如图 15

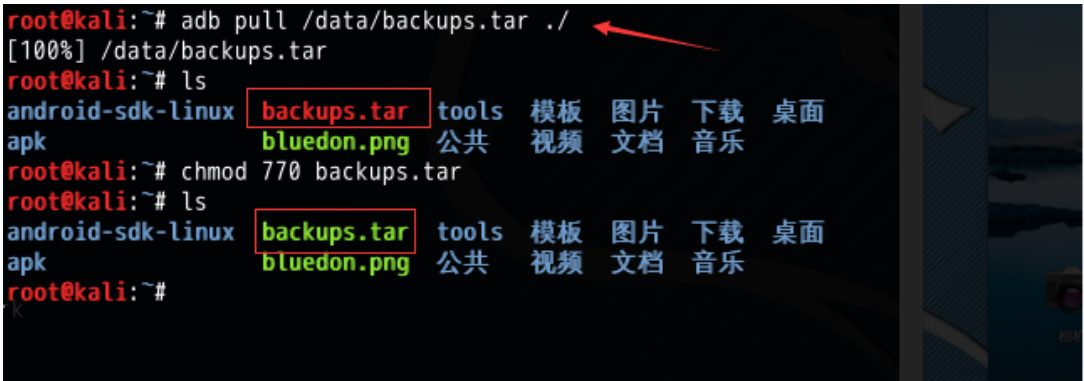


图 15备份下载到kali

16、解压数据库备份文件。

操作：“tar -zxvf backups.tar”如图 16

```
root@kali:~# tar -zxvf backups.tar
dbbackup/
dbbackup/grants.db
dbbackup/EmailProvider.db
dbbackup/EmailProviderBackup.db
dbbackup/EmailProviderBody.db
dbbackup/WebpageIcons.db
dbbackup/browser2.db
dbbackup/webview.db
dbbackup/webviewCookiesChromiumPrivate.db
dbbackup/webviewCookiesChromium.db
dbbackup/autofill.db
dbbackup/Databases.db
dbbackup/ApplicationCache.db
dbbackup/CachedGeoposition.db
dbbackup/GeolocationPermissions.db
dbbackup/drozer.db
dbbackup/qsba-log.db
dbbackup/internal.db
dbbackup/alarms.db
dbbackup/downloads.db
```

图 16解压缩

17、进入到dbbackup目录查看是否解压成功。

操作：“cd dbbackup”>“ls”如图 17

```
root@kali:~# cd dbbackup/
root@kali:~/dbbackup# ls
alarms.db             grants.db
ApplicationCache.db   internal.db
autofill.db           launcher.db
browser2.db           mmssms.db
CachedGeoposition.db  profile.db
calendar.db           qsb-log.db
contacts2.db          settings.db
Databases.db          telephony.db
downloads.db          userbigram_dict.db
drozer.db             user_dict.db
EmailProviderBackup.db WebpageIcons.db
EmailProviderBody.db  webviewCookiesChromium.db
EmailProvider.db      webviewCookiesChromiumPrivate.db
GeolocationPermissions.db webview.db
root@kali:~/dbbackup#
```

图 17解压成功

18、打开sqlitebrowser。

操作：“在终端输入sqlitebrowser命令”，如图 18



图 18 sqlitebrowser界面

19、打开数据库文件，并且查看相应的数据库表。

操作：“单击打开数据库”>“选取数据库文件所在的目录” 如图 19图 20

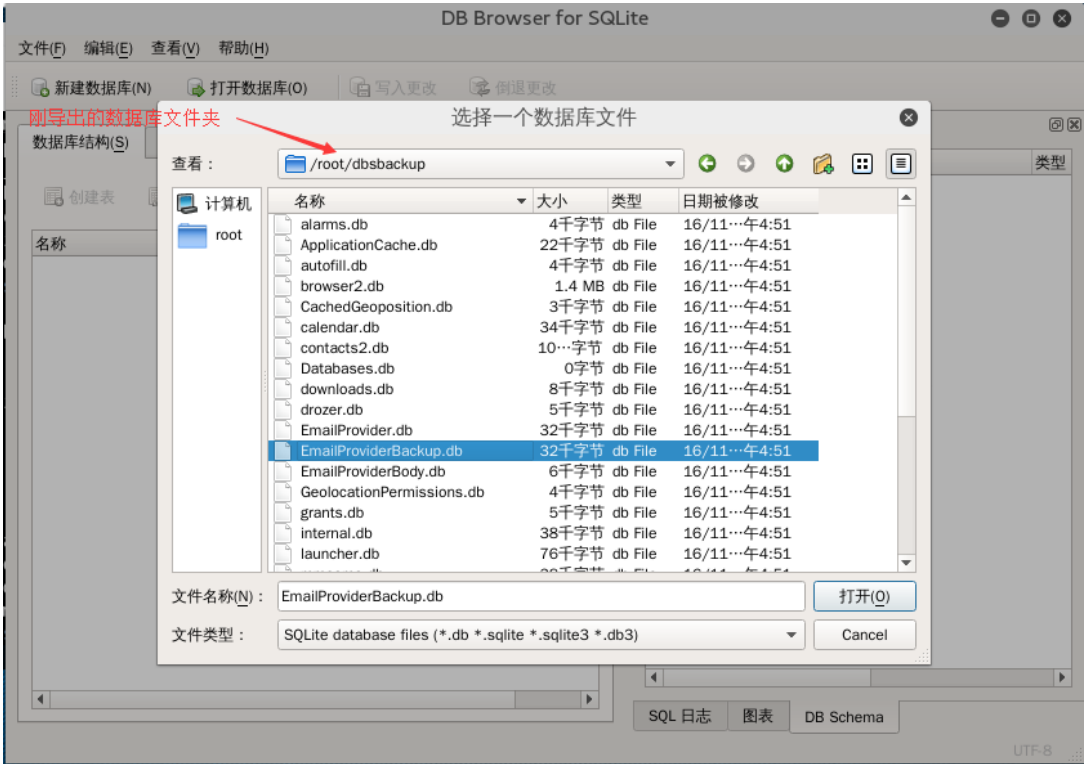


图 19打开数据库文件

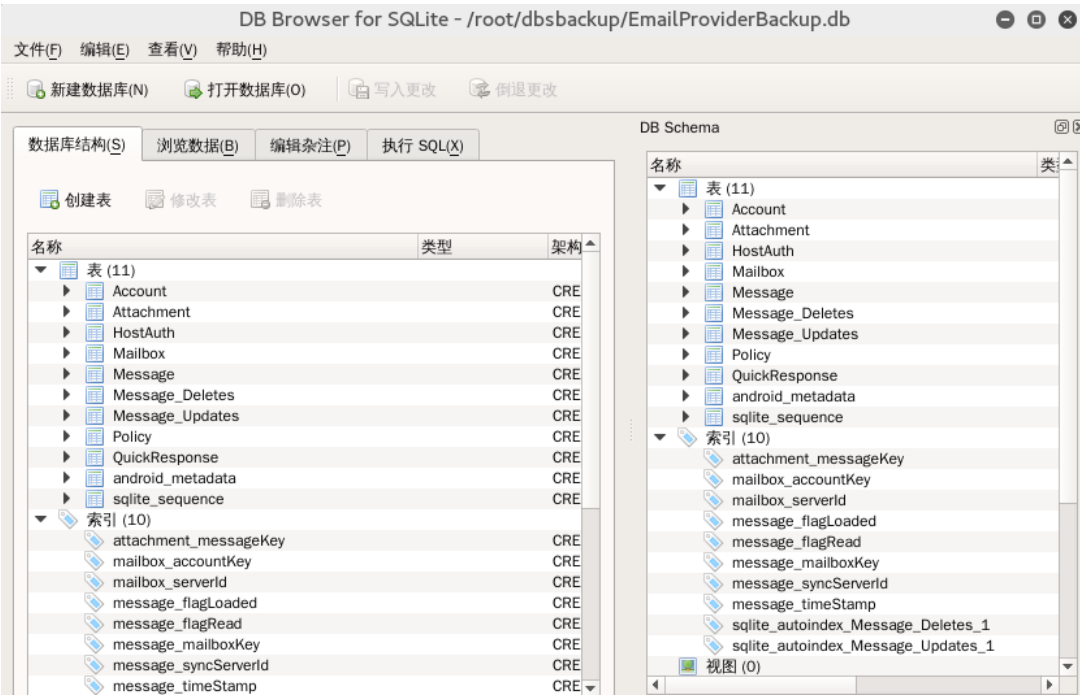


图 20查看数据库表

思考总结

本实验通过安装busybox来使用find、tar等linux命令，首先获取一个android shell 把数据库文件找出并且进行备份，然后把备份的数据库文件传输到电脑，最后使用sqlitebrowser对数据库文件进行查看，验证是否备份成功。

- 1、为什么要对数据库文件进行备份？
- 2、busybox的功能是什么？