

安卓上Web漏洞的自动化检测

• Fooying

• 一句话其实是搞Web安全的程序猿 → → 屌丝

• 欢迎关注我的公众号 → → → → → → → →



- PC上常规的漏洞挖掘与自动化检测
- 安卓APP的服务端请求
- 手动挖掘安卓APP里的Web漏洞
- 实现安卓上Web漏洞挖掘的自动化
- 安卓上应用静态分析挖掘Web漏洞与对比
- 更多的自动化检测

PC上常规的漏洞挖掘与自动化检测

- 常见Web漏洞

- SQLI、XSS、CSRF、命令执行... (OWASP TOP 10)

- 漏洞产生

- 对非预期输入的信任

• 常规Web漏洞检测方式

- 选择目标, 针对性或者非针对性目标页面
- Payload, 根据漏洞类型构造漏洞测试语句
- 尝试提交
- 根据响应判断结果



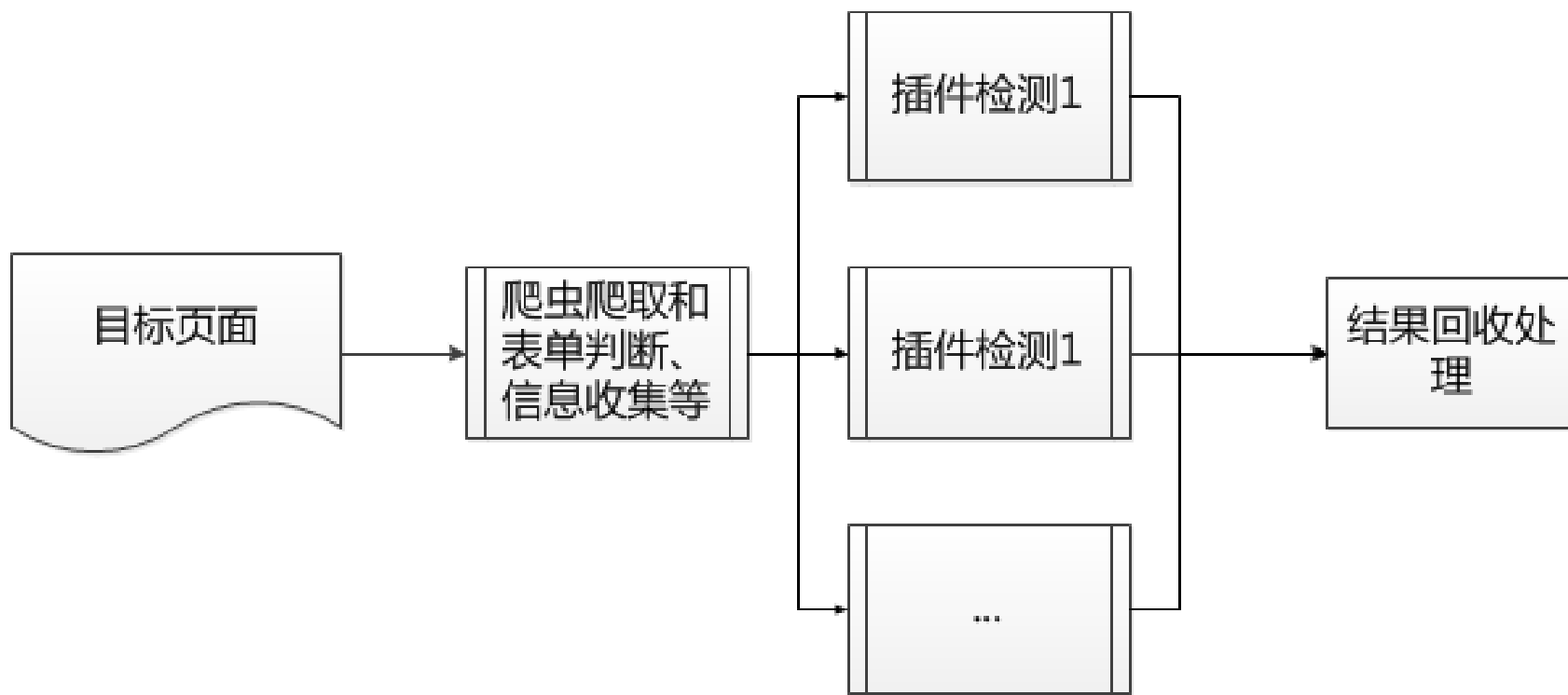
- 如XSS的测试

- 页面存在输入点，提交内容存在于页面
- 构造payload，如
- 尝试提交
- 页面是否弹框

针对向量（输入点）的Fuzzing

· 扫描器的组成

- 爬虫
- 漏洞检测插件
- 调度、结果回收等框架功能



安卓APP的服务端请求

- “Web型的应用”



• Web 请求操作

— 请求线上的API

- 请求配置文件
- 登录、注册等
- 同步本地数据
- ...

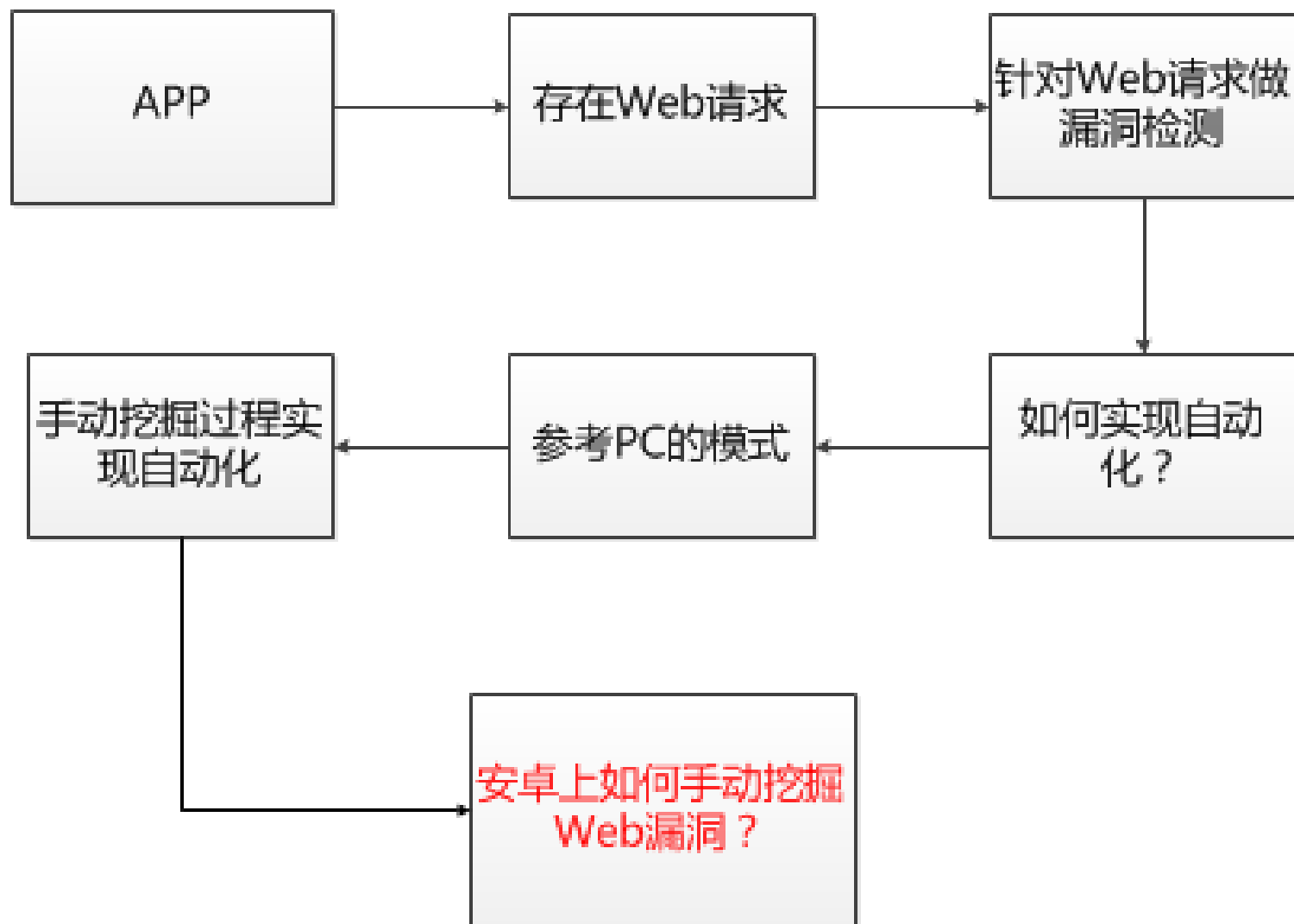
— 内嵌Web服务

- 注册、登录页面
- 页面移动版
- 引用地图等
- ...



手动挖掘安卓APP里的Web漏洞

· 思考流程



- 对比PC端的Web漏洞挖掘问题？

- 孤岛URL
- 无法控制向量进行Fuzzing!
- 无法使用各种辅助工具!
- ...

— 有些难度

- 思路转换

- 可否将安卓上Web漏洞挖掘在PC平台上进行?
- 但如何获取这些URLs?

- 静态分析

- 反编译APK文件
- 审计源码
- 寻找请求的URL

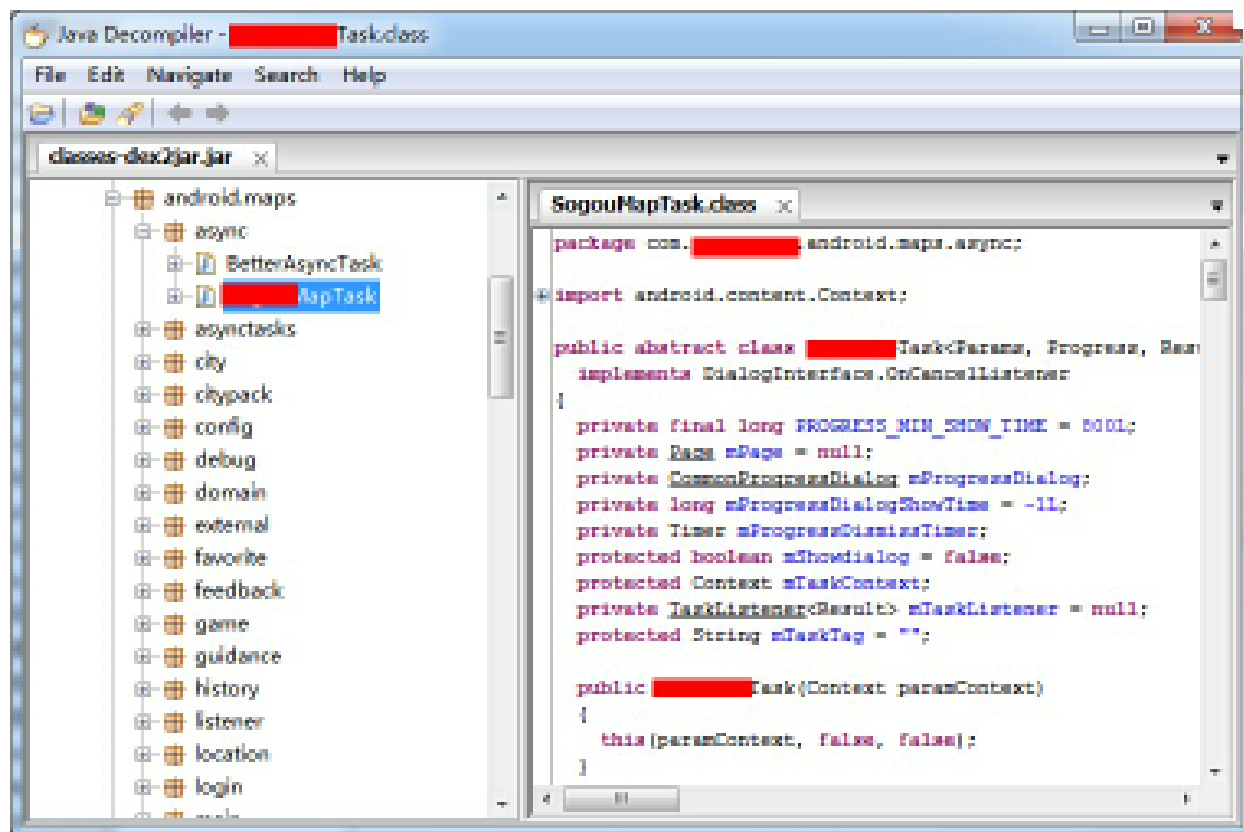
• Dex2jar&jd-gui

- unzip simple.apk
- Dex2jar classes.dex → jar file
- jd-gui → view jar file

```
./dex2jar.sh /vagrant/[REDACTED]-gphone-last/classes.dex
```

名称

- assets
- lib
- META-INF
- res
- AndroidManifest.xml
- classes.dex
- resources.arsc



• Androguard搜索URL

- `./androlyze.py -s`
- `a, d, dx = AnalyzeAPK('simple.apk', decompiler='dad')`
- `d.get_regex_strings('.*http://.*')`
- `u = dx.tainted_variables.get_string(url)`
- `u.show_paths(d)`

```
vagrant@packer-virtualbox:/vagrant/fooying/f/tools/androguard$ ./androlyze.py -s
/usr/local/lib/python2.7/dist-packages/IPython/frontend.py:30: UserWarning: The top-level `frontend` package has been
subpackages have been moved to the top `IPython` level.
  warn("The top-level `frontend` package has been deprecated. ")
Androlyze version 2.0
In [1]: a,d,dx = AnalyzeAPK('/vagrant/other/weibo 10105011.apk', decompiler='dad')

In [2]: d.get_regex_strings('.*http://.*')
Out[2]:
['http://3g.sina.com.cn/interface/f/ttt/v2/',
'http://3g.sina.com.cn/interface/f/ttt/v2/login.php']

In [3]: u = dx.tainted_variables.get_string('http://3g.sina.com.cn/interface/f/ttt/v2/login.php')

In [4]: u.show_paths(d)
R 16a Lcom/sina/weiboapp/RPCHelper;->login (Ljava/lang/String; Ljava/lang/String;)Lcom/sina/weiboapp/models/User;

In [5]: d.CLASS Lcom.sina.weiboapp.RPCHelper.METHOD login.source()
```

• Androguard搜索URL

```
In [5]: d.CLASS_Lcom_sina_weiboapp_RPCHelper.METHOD_login.source()

public static com.sina.weiboapp.models.User login(String p14, String p15)
{
    char[] v7 = com.sina.weiboapp.MD5.hexdigest(new StringBuilder(String.valueOf(p14)).append(p15).append("510WXnhiY4pJ794KIJ7Rw5F45V
Xg9sjo").toString()).toCharArray();
    StringBuffer v4_2 = new StringBuffer().append(v7[1]).append(v7[5]).append(v7[2]).append(v7[10]).append(v7[17]).append(v7[9]).appe
nd(v7[25]).append(v7[27]);
    java.util.ArrayList v3_1 = new java.util.ArrayList();
    v3_1.add(new org.apache.http.message.BasicNameValuePair("u", p14));
    v3_1.add(new org.apache.http.message.BasicNameValuePair("p", p15));
    v3_1.add(new org.apache.http.message.BasicNameValuePair("c", "android"));
    v3_1.add(new org.apache.http.message.BasicNameValuePair("s", v4_2.toString()));
    v3_1.add(new org.apache.http.message.BasicNameValuePair("from", "10105011"));
    v3_1.add(new org.apache.http.message.BasicNameValuePair("wm", ""));
    try {
        org.apache.http.client.entity.UrlEncodedFormEntity v2_1 = new org.apache.http.client.entity.UrlEncodedFormEntity(v3_1, "UTF-8
");
        org.apache.http.client.methods.HttpPost v6_1 = new org.apache.http.client.methods.HttpPost("http://3g.sina.com.cn/interface/f
/ttt/v2/login.php");
        v6_1.setEntity(v2_1);
        String v0 = com.sina.weiboapp.RPCHelper.execute(v6_1);
        com.sina.weiboapp.models.User v9_1 = new com.sina.weiboapp.models.User();
        org.xmlpull.v1.XmlPullParser v5 = android.util.Xml.newPullParser();
    } catch (String v10_28) {
        throw new com.sina.weiboapp.RPCHelper$ApiException(com.sina.weiboapp.RPCHelper.UNKNOWN_ERROR, v10_28);
    }
}
```

• Apktool

— `java -jar apktool.jar d simple.apk`

```

java -jar apktool1.5.2.jar d /vagrant/other/weibo_10105011.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/vagrant/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...

```

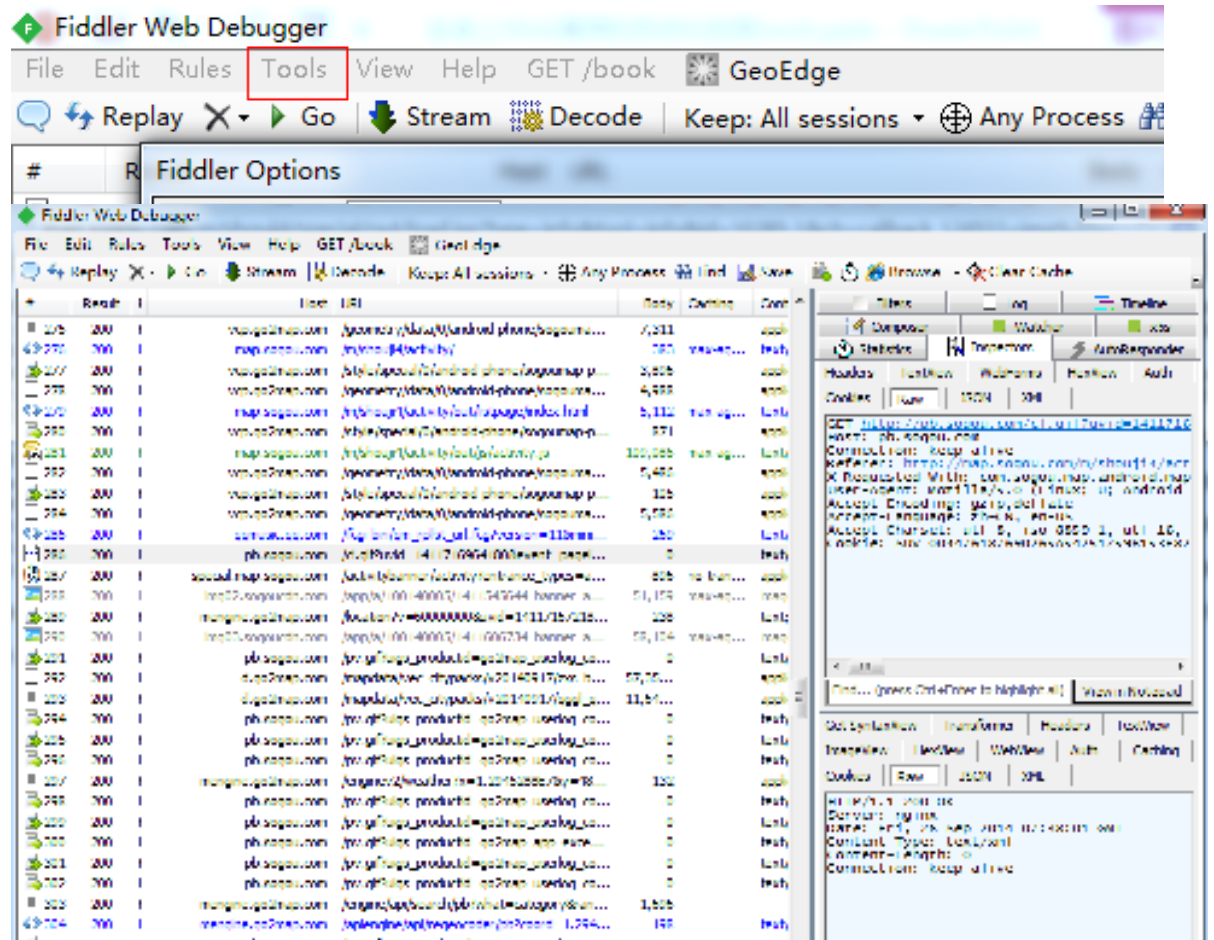
```

.
|-- AndroidManifest.xml ← 反编译过的配置文件
|-- apktool.yml
|-- res
|   |-- drawable
|   |-- layout
|   |-- menu
|   |-- values
|   |-- values-en
|   |-- values-zh-rHK
|   `-- values-zh-rTW ← smali源码目录
`-- smali ←
    |-- com

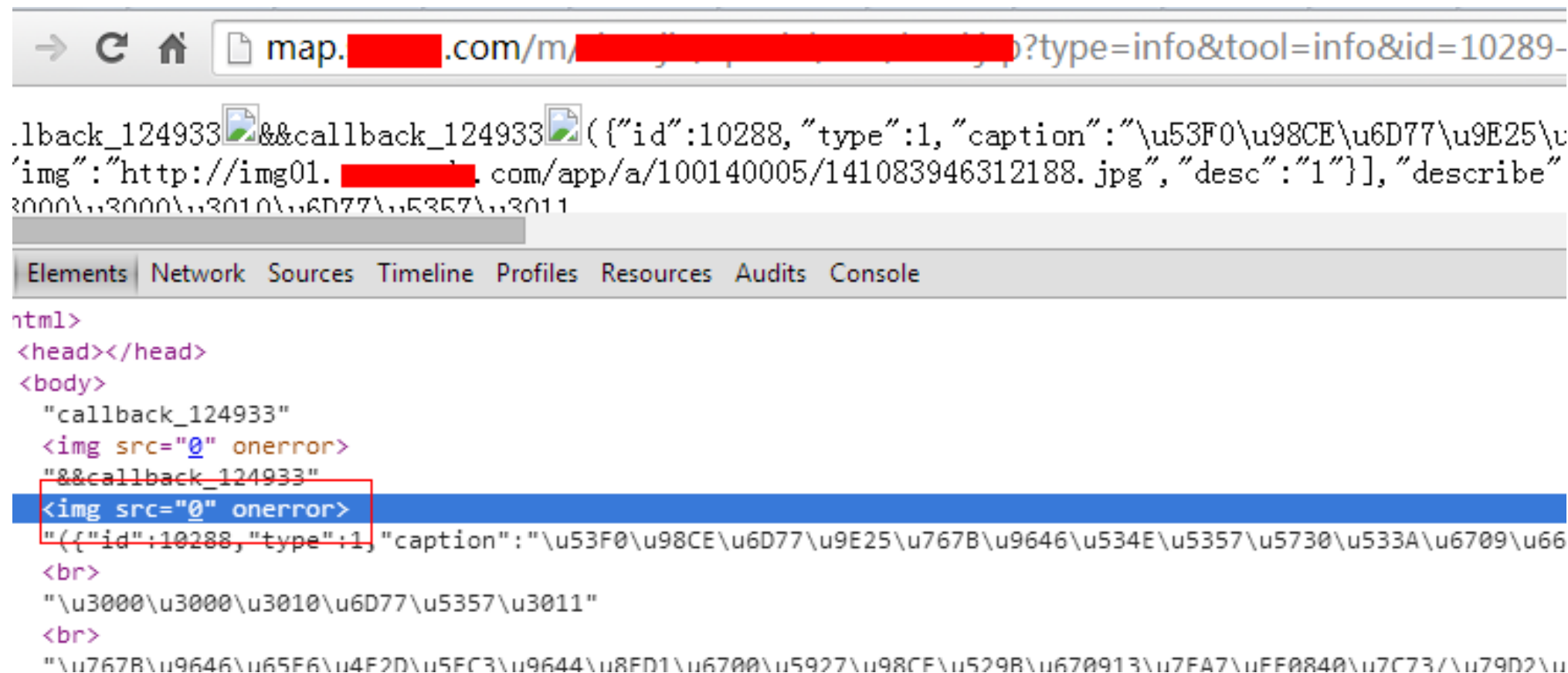
```

• 代理捕获请求URL

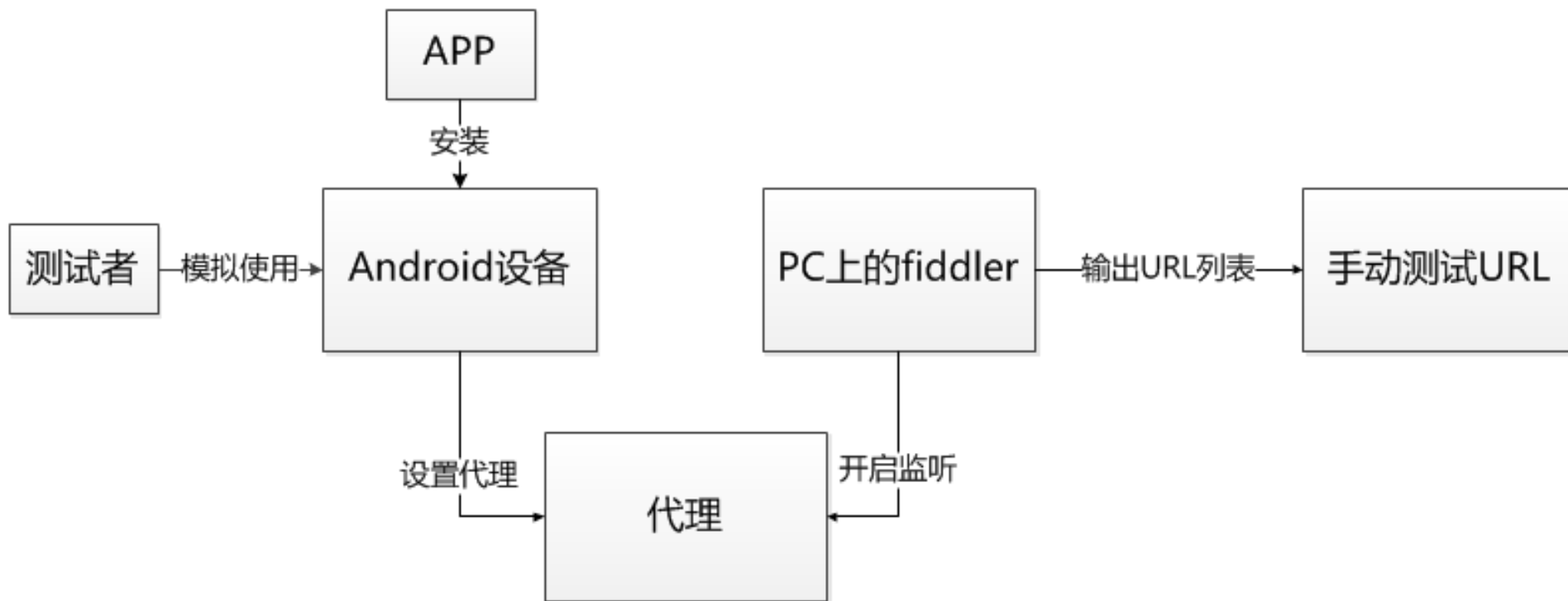
- 通过fiddler在pc端开启代理
- 将手机与PC置于同一局域网，并设置代理
- 通过fiddler捕获请求



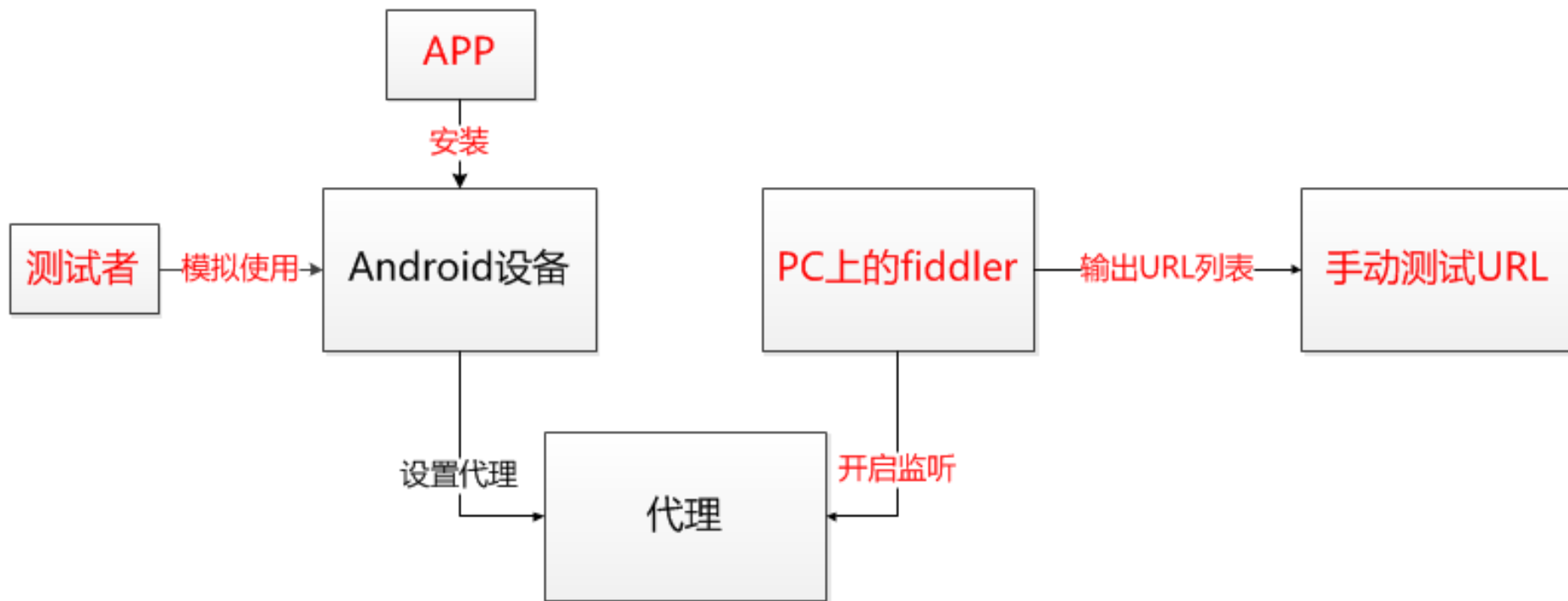
· 漏洞示例



· 手动测试流程



- 可以自动化的环节



- 主要要处理的部分

- 1、 安卓模拟器及代理设置
- 2、 APP的安装与自动卸载
- 3、 模拟执行操作应用功能触发请求
- 4、 开启代理监听请求
- 5、 取请求列表进行自动化漏洞检测

• APK 文件

```
.
| -- AndroidManifest.xml
| -- assets
| -- classes.dex
| -- lib
| -- META-INF
| -- res
\ -- resources.arsc
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="60000000"
    android:versionName="6.0.0"
    android:installLocation="0"
    package="com.sogou.map.android.maps"
>
    <application
        android:theme="@7F0B0031"
        android:label="@7F060002"
        android:icon="@7F0201C0"
        android:name=".SogouMapApplication"
        android:debuggable="false"
    >
        <activity
            android:theme="@7F0B0033"
            android:label="@7F060002"
            android:name=".SplashActivity"
            android:screenOrientation="1"
        >
            <intent-filter
                >
                    <action
                        android:name="android.intent.action.MAIN"
                    >
                </action>
                <category
                    android:name="android.intent.category.LAUNCHER"
                >
            </intent-filter>
        </activity>
    </application>
</manifest>
```

· 1、安卓模拟器及代理设置

- Android模拟器

- Root Android模拟器

- 使用ProxyDroid设置全局代理

- Android-x86虚拟机镜像下载 <http://www.android-x86.org/download>

· 2、APP的安装与自动卸载

- 安装 `adb install apk_file`
- 卸载 `adb uninstall pkg_name`
- `apk_file` eg. `e:\aaa.apk`
- `pkg_name` eg. `com.apps.demo`

· 3、模拟执行操作应用功能触发请求

- Activity

· <https://developer.android.com/guide/topics/manifest/activity-element.html>

DESCRIPTION:

Declares an activity (an `Activity` subclass) that implements part of the application's visual user interface. All activities must be represented by `<activity>` elements in the manifest file. Any that are not declared there will not be seen by the system and will never be run.

- 声明一个活动（一个Activity子类），实现了应用程序的可视化用户界面的一部分。所有的活动必须在manifest文件中以`<activity>`表示。任何未声明的活动不会被系统识别并且将不会执行。

· 3、模拟执行操作应用功能触发请求

- 解压apk文件 (unzip example.apk)
- java -jar AXMLPrinter2.jar AndroidManifest.xml >newxml.xml
- Activity的遍历与触发


```
java -jar AXMLPrinter2.jar AndroidManifest.xml > newxml.xml
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:versionCode="60000000"
  android:versionName="6.0.0"
  android:installLocation="0"
  package="com.sogou.map.android.maps"
>
  <application
    android:theme="@7F0B0031"
    android:label="@7F060002"
    android:icon="@7F0201C0"
    android:name=".SogouMapApplication"
    android:debuggable="false"
  >
    <activity
      android:theme="@7F0B0033"
      android:label="@7F060002"
      android:name=".SplashActivity"
```

• Activity的遍历与触发

- start an Activity: `am start [-D] [-W] <INTENT>`
 - D: enable debugging
 - W: wait for launch to complete
- 读取xml文件, 读取节点的值 `activity_name`
- `adb shell am start -n pkg_name/activity_name`
- `adb shell ps | grep %s | awk '{print $2}' | xargs %s shell kill`

• Activity的遍历与触发

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="60000000"
    android:versionName="6.0.0"
    android:installLocation="0"
    package="com.sogou.map.android.maps"
    >
    <application
        android:theme="@7F0B0031"
        android:label="@7F060002"
        android:icon="@7F0201C0"
        android:name=".SogouMapApplication"
        android:debuggable="false"
        >
        <activity
            android:theme="@7F0B0033"
            android:label="@7F060002"
            android:name=".SplashActivity"
            android:screenOrientation="1"
            >
            <intent-filter
                >
                <action
                    android:name="android.intent.action.MAIN"
                    >
                </action>
                <category
                    android:name="android.intent.category.LAUNCHER"
                    >
                </category>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

← pkg_name

← activity_name

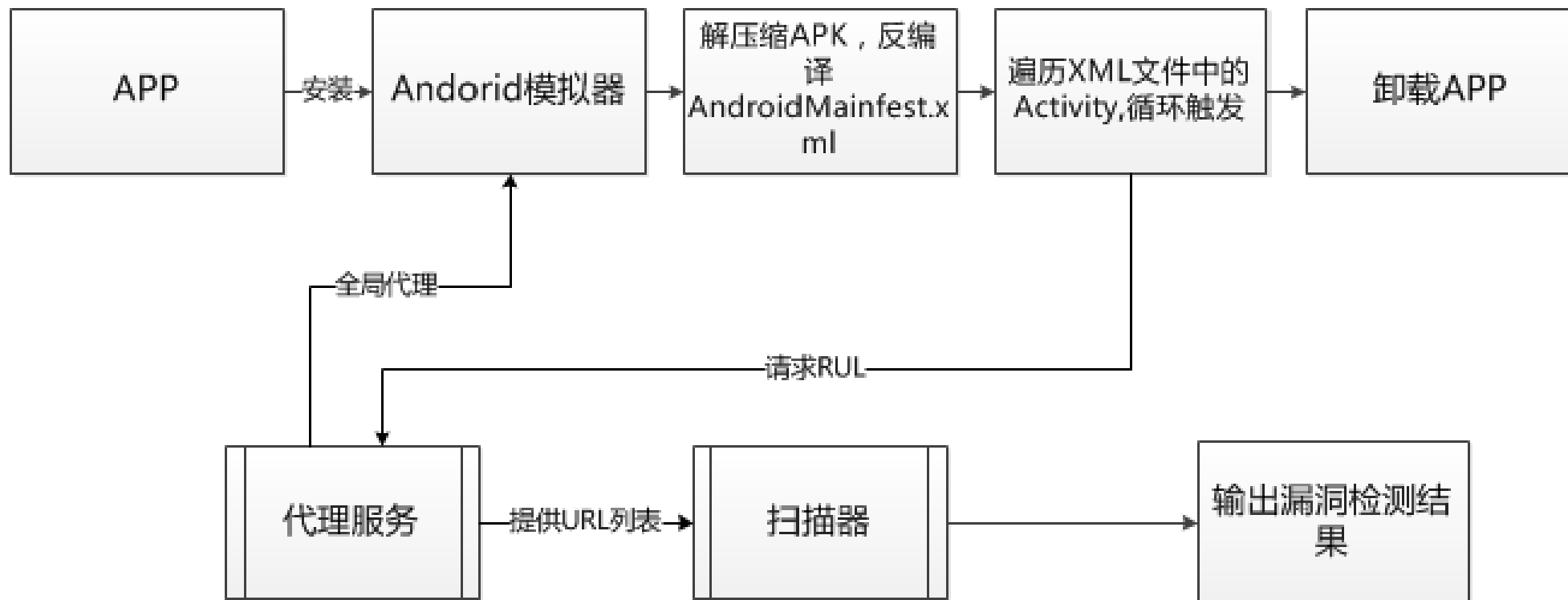
- 4、开启代理监听请求
- 5、取请求列表进行自动化漏洞检测

- 自开发的扫描器或者脚本等
- 通过burpsuite等
- ...

· 关键部分流程

- 1、APP的安装与自动卸载
- 2、AndroidManifest.xml文件的反编译
- 3、Activity的遍历与触发
- 4、代理收集请求
- 5、漏洞的检测

· 检测流程



- 还有一些问题

- https 的问题

- 提取证书安装?
 - Hook证书验证代码?
 - 模拟器 iptables 端口映射?

- 模拟器

- 如何方便的批量部署模拟器?
 - 部分app安装闪退
 - 应用奔溃?

- ...

- 实际检测演示

- 静态分析处理过程

- 1、反编译apk包里的相关文件
- 2、扫描匹配反编译的文件查找URL
- 3、针对URL进行检测

· 动态分析VS静态分析

	静态分析	动态分析
优点	不需要模拟器	完整获取请求
	可以忽略是否https请求	获取大部分可能发生的URL
缺点	获取的URL不完整	需要模拟器
	安全加固的apk反编译困难	代理无法直接获取https请求

• 待拼接的URL

```
public static boolean deleteFavMblog(User paramUser, String paramString)
    throws RPCHelper.ApiException, RPCHelper.ParseException
{
    Object[] arrayOfObject = new Object[8];
    arrayOfObject[0] = "http://3g. .com.cn/interface/f/ttt/v2/";
    arrayOfObject[1] = paramUser.sid;
    arrayOfObject[2] = paramUser.gsid;
    arrayOfObject[3] = paramString;
    arrayOfObject[4] = "android";
    arrayOfObject[5] = calculateS(paramUser.uid);
    arrayOfObject[6] = "10105011";
    arrayOfObject[7] = "";
    return parseResult(execute(new HttpGet(String.format(
        "%sdealfavmblog.php?sid=%s&gsid=%s&act=1&id=%s&c=%s&s=%s&from=%s&wm=%s", arrayOfObject))));
}
```

- 也许只得到http://3g.xxx.com.cn/interface/f/ttt/v2/
- 一些变量的值无从得知

- 微信公众号
- 各种后端请求的服务

· 微信公

一 开发

· 同

· 所以



IOS上的应用是否可以同理可现？

[←](#)
[→](#)
[↺](#)

m.25pp.com/ios_apple/#appDetail

[<返回](#)

简介



语言: 中文

★★★★★

查火车票 - 12306

版本: 2.5.1	下载: 21032
分类: 商品指南	大小: 2.56M
开发商: tang jiaping	
系统要求: 需要 iOS 4.3 或更高版本。与 iPhone、iPad、iPod touch 兼容。此 App 已针对 iPhone 5 优化。	

立即安装

立即下载

升级日志

- 1、适配iOS8
- 2、增加车次查询功能

· ipa 应用测试截图

```
heigemato-iPad:/var/mobile root# python ./ipa.py http://m.25pp.com/ios_apple/plist/556789911.plist
[+]Download http://zbsoft.25pp.com/share/151/556789911_1412758362.ipa...
--2014-10-10 19:35:39-- http://zbsoft.25pp.com/share/151/556789911_1412758362.ipa
Resolving zbsoft.25pp.com... 117.41.175.200, 117.41.175.196, 117.41.175.207, ...
Connecting to zbsoft.25pp.com|117.41.175.200|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2688928 (2.6M) [application/octet-stream]
Saving to: `./down.ipa'

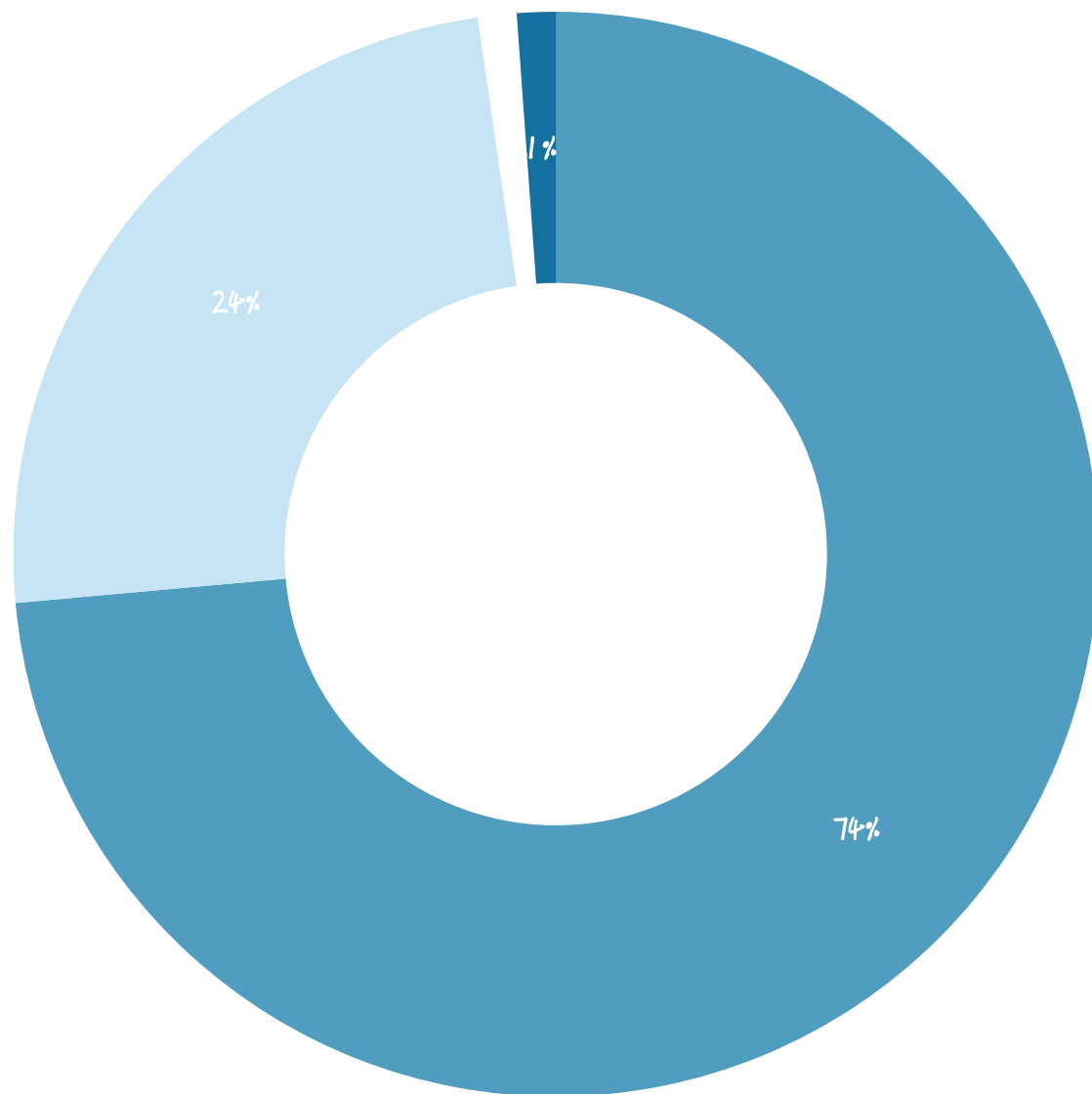
100%[=====>] 2,688,928    269K/s   in 9.7s

2014-10-10 19:35:53 (270 KB/s) - `./down.ipa' saved [2688928/2688928]

[+]Install http://zbsoft.25pp.com/share/151/556789911_1412758362.ipa...
2014-10-10 19:36:05.094 heige[2900:c07] Installed.
[+]Run http://zbsoft.25pp.com/share/151/556789911_1412758362.ipa...
[+]Rm local ipa file
```


各种漏洞占比

■ XSS
 ■ SQL注入
 ■ Struts任意代码执行
 ■ 本地任意文件读取



- <http://ctl.ifeng.com/>* 凤凰网
- <http://i.meituan.com/>* 美团
- <http://t.bypay.cn/>* 百付天下
- <https://client.bestpay.com.cn/>* 翼支付
- <http://icar.qq.com/>* 腾讯
- <http://tips.passport.pptv.com/>* PPTV
- <http://bbx2.sj.91.com/>* 91
- <http://help.pcl20.com/>* 金山
- <http://mm.maxthon.cn/>* 遨游
- ...

- Fooying

- 邮箱: `f00y1n9@gmail.com`
- 微博: `@cnfooying`
- 博客: www.fooying.com



谢谢