漏洞类型	高危
漏洞代码所在包名	com.avos.sns
漏洞代码所在类名	SNSWebActivity
漏洞代码所在方法名	onCreate
漏洞代码	myWebView.addJavascriptInterface(new AVSNSWebViewCallback(th is), "snsCallback");
漏洞描述	调用了高危API addJavascriptInterface,Android系统版本4.2以下可以导致远程命令执行
修复建议	修复建议：不使用addJavascriptInterface进行 javascript层与java层的通信，可以使用shouldOverrideUrlLoading或者onConsoleMessage等函数进行通信

漏洞类型	高危
漏洞代码所在包名	com.doyutown.fishpond.ui
漏洞代码所在类名	WebDetailActivity
漏洞代码所在方法名	setWebViewListener
漏洞代码	this.mWebView.addJavascriptInterface(this.mWebAppInterface, "AndroidWebAppInterface");
漏洞描述	调用了高危API addJavascriptInterface,Android系统版本4.2以下可以导致远程命令执行
修复建议	修复建议：不使用addJavascriptInterface进行 javascript层与java层的通信，可以使用shouldOverrideUrlLoading或者onConsoleMessage等函数进行通信

漏洞类型	中危
漏洞代码所在包名	android.support.annotation
漏洞代码	android:allowBackUp=true
漏洞描述	不要在发行版本中设置android:allowbackup='true'
修复建议	修复建议:不要在发行版本中设置android:allowbackup='true'

漏洞类型	中危
漏洞代码所在包名	com.google.android.gms.auth
漏洞代码所在类名	GoogleAuthUtil

漏洞代码所在方法名	zzi
漏洞代码	Intent.parseUri(intent.toUri(CHANGE_TYPE_ACCOUNT_ADDED), CHANGE_TYPE_ACCOUNT_ADDED);
漏洞描述	使用Intent.parseUri时，需要确保参数来源安全,如果参数外部可控，那么恶意攻击者可以通过构造恶意uri 直接访问未公开组件和私有文件
修复建议	修复建议：确保外部无法控制uri参数，如果可控，加上三行：intent.addCategory('android.intent.category.BROWSABLE');intent.setComponent(null); intent.setSelector(null);

漏洞类型	中危
漏洞代码所在包名	com.alibaba.sdk.android.webview
漏洞代码所在类名	BaseWebViewClient
漏洞代码所在方法名	onReceivedSslError
漏洞代码	sslErrorHandler.proceed();
漏洞描述	在webview打开证书有问题的https网页时，在onreceivedsslerror回调中直接忽略错误,则会导致可以通过中间人攻击的方式劫持https流量
修复建议	修复建议：不要重载onReceivedSslError函数 ，如果需要重载onReceivedSslError函数，那么 不要调用SslErrorHandler#proceed() 函数忽略错误

漏洞类型	中危
漏洞代码所在包名	com.alibaba.sdk.android.webview
漏洞代码所在类名	BaseWebViewClient
漏洞代码所在方法名	onReceivedSslError
漏洞代码	sslErrorHandler.proceed();
漏洞描述	在webview打开证书有问题的https网页时，在onreceivedsslerror回调中直接忽略错误,则会导致可以通过中间人攻击的方式劫持https流量
修复建议	修复建议：不要重载onReceivedSslError函数 ，如果需要重载onReceivedSslError函数，那么 不要调用SslErrorHandler#proceed() 函

	数忽略错误

漏洞类型	中危
漏洞代码所在包名	com.tencent.open.utils
漏洞代码所在类名	HttpUtils$MyX509TrustManager
漏洞代码所在方法名	checkServerTrusted

漏洞代码	public void checkServerTrusted(X509Certificate[] x509Certificate, String str) throws CertificateException {
	
	try {
	this.a.checkServerTrusted(x509CertificateArr, str);
	} catch (CertificateException e) {
	}
	}

漏洞描述	如果自己实现https证书管理manager，那么checkServerTrusted方法不能实现为空,会导致黑客可以通过中间人攻击的方式劫持https流量
修复建议	修复建议:实现checkServerTrusted或者调用父类函数

漏洞类型	中危
漏洞代码所在包名	com.loopj.android.http
漏洞代码所在类名	MySSLSocketFactory
漏洞代码所在方法名	checkServerTrusted
漏洞代码	public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException { }
漏洞描述	如果自己实现https证书管理manager，那么checkServerTrusted方法不能实现为空,会导致黑客可以通过中间人攻击的方式劫持https流量
修复建议	修复建议:实现checkServerTrusted或者调用父类函数

漏洞类型	中危
漏洞代码所在包名	com.loopj.android.http
漏洞代码所在类名	MySSLSocketFactory
漏洞代码所在方法名	getFixedSocketFactory

漏洞代码	SSLSocketFactory can not setHostnameVerifier with ALLOW_ALL_HOSTNAME_VERIFIER
漏洞描述	调用setHostnameVerifier(ALLOW_ALL_HOSTNAME_VERIFIER) 会信任证书错误的响应包,会导致黑客可以通过中间人攻击的方式劫持https流量
修复建议	修复建议：setHostnameVerifier(SSLSocketFactory.STRICT_HOSTNAME_VERIFIER);

漏洞类型	中危
漏洞代码所在包名	org.jsoup.helper
漏洞代码所在类名	HttpConnection$Response
漏洞代码所在方法名	checkServerTrusted
漏洞代码	public void checkServerTrusted(X509Certificate[] chain, String authType) { }
漏洞描述	如果自己实现https证书管理manager, 那么checkServerTrusted方法不能实现为空,会导致黑客可以通过中间人攻击的方式劫持https流量
修复建议	修复建议:实现checkServerTrusted或者调用父类函数

漏洞类型	中危
漏洞代码所在包名	com.umeng.socialize.view
漏洞代码所在类名	a
漏洞代码所在方法名	onReceivedSslError
漏洞代码	sslErrorHandler.proceed();
漏洞描述	在webview打开证书有问题的https网页时, 在onreceivedsslerror回调中直接忽略错误,则会导致可以通过中间人攻击的方式劫持https流量
修复建议	修复建议: 不要重载onReceivedSslError函数 , 如果需要重载onReceivedSslError函数, 那么 不要调用SslErrorHandler#proceed() 函数忽略错误

漏洞类型	低危
漏洞代码所在包名	com.avos.avoscloud
漏洞代码所在类名	SessionManager
漏洞代码所在方法名	sendErrorBroadcast

漏洞代码	AVOSCloud.applicationContext.sendBroadcast(exceptionIntent);
漏洞描述	使用隐式Intent(即未指定目标组件名称)来发送广播,恶意程序可以通过注册收听同样action的broadcastreceiver来劫持Intent, 如果Intent中有敏感信息, 那么会造成敏感信息的泄露.
修复建议	修复建议:如果Intent中包含敏感信息, 那么在sendBroadCast之前需要显式指定component

漏洞类型	低危
漏洞代码所在包名	com.tencent.mm.sdk.a.a
漏洞代码所在类名	a
漏洞代码所在方法名	a
漏洞代码	context.sendBroadcast(intent, str);
漏洞描述	使用隐式Intent(即未指定目标组件名称)来发送广播,恶意程序可以通过注册收听同样action的broadcastreceiver来劫持Intent, 如果Intent中有敏感信息, 那么会造成敏感信息的泄露.
修复建议	修复建议:如果Intent中包含敏感信息, 那么在sendBroadCast之前需要显式指定component

漏洞类型	低危
漏洞代码所在包名	com.avos.avoscloud
漏洞代码所在类名	BroadcastUtil
漏洞代码所在方法名	sendSessionBroadCast
漏洞代码	AVOSCloud.applicationContext.sendBroadcast(sessionIntent);
漏洞描述	使用隐式Intent(即未指定目标组件名称)来发送广播,恶意程序可以通过注册收听同样action的broadcastreceiver来劫持Intent, 如果Intent中有敏感信息, 那么会造成敏感信息的泄露.
修复建议	修复建议:如果Intent中包含敏感信息, 那么在sendBroadCast之前需要显式指定component

漏洞类型	低危
漏洞代码所在包名	com.tencent.wxop.stat
漏洞代码所在类名	g
漏洞代码所在方法名	aa

漏洞代码	this.bh.getApplicationContext().registerReceiver(new z(this), new IntentFilter("android.net.conn.CONNECTIVITY_CHANGE"));
漏洞描述	使用.registerReceiver(mReceiver, intentFilter)的方式动态注册的BroadCastReceiver,是公开的组件, 外部应用可以给该动态注册的BroadCastReceiver发送恶意数据
修复建议	建议使用registerReceiver(BroadcastReceiver receiver, IntentFilter filter, String broadcastPermission, Handler scheduler)增加权限或者使用LocalBroadCastManager