

# 基于源代码的漏洞挖掘

曹宗伟  
ayound

主办方

联合主办方

安全+



WiFi万能钥匙安全应急响应中心  
WiFi MasterKey Security Response Center



苏宁安全应急响应中心  
Suning Security Response Center

协办方



ALIBABA SECURITY  
RESPONSE CENTER



eWent  
安全应急响应中心



平安安全应急响应中心  
PINGAN Security Response Center



网易安全应急响应中心  
NetEase Security Response Center



搜财安全应急响应中心  
Wealth Security Response Center



安全城市  
SAFE CITY



iQIYI  
ZISRC



唯品会安全应急响应中心  
Veeva Security Response Center



美丽联合安全  
Meili Alliance Security



微博安全应急响应中心  
Weibo Security Response Center



安全加

# 基于源代码的漏洞挖掘

JAVA

漏洞分析思路 and 工具

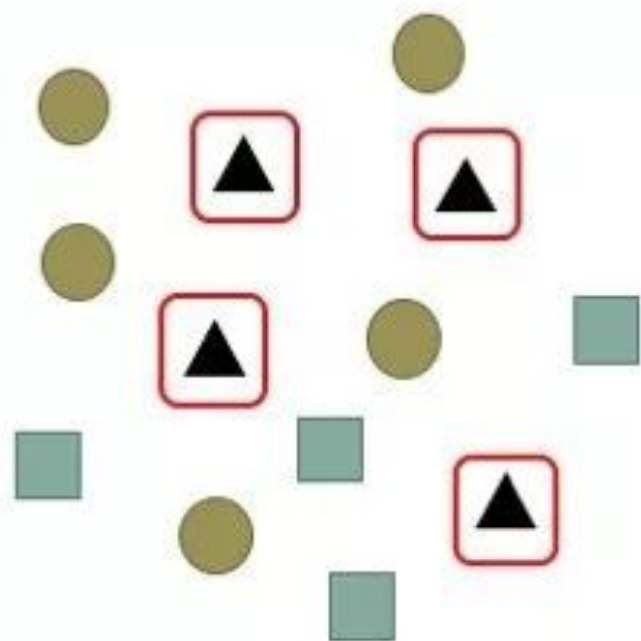
未公开的两个POC

安卓App漏洞分析



# 核心思路→基于代码路径的分析

## 基于文本和规则的扫描

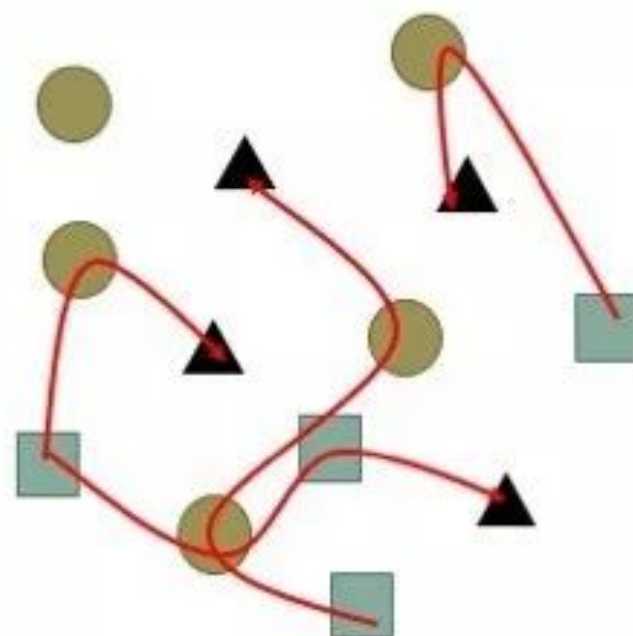


■ 攻击面

▲ 问题代码



## 基于代码路径的分析



● 正常代码

安全

安全加

# 基于路径分析的核心

## 基于IDE

- 代码
- 环境
- 依赖

## 索引

- callee
- caller
- extend
- implement

## 攻击面识别

- 反序列化
- 安卓

## 搜索

- 正向搜索
- 反向搜索



# 为什么基于IDE？



# 为什么基于IDE?



找出所有interface1的实现类



# 为什么需要继承或接口实现索引？

## 以JSON反序列化所需要的gadget为例



安全

# 为什么需要继承或接口实现索引？

```
Thread thread = new TestThread(abc);  
thread.start();
```



```
TestThread的方法  
public void run() {  
    //漏洞代码  
    ...  
}
```



# 为什么需要继承或接口实现索引？

```
Intent intent = new Intent(Intent.ACTION_VIEW,  
TestActivity.class);  
startActivity(intent);
```



## TestActivity的方法

```
public void onCreate(Intent intent) {  
    //漏洞代码  
}
```

# 攻击面识别→以json反序列化为例



...

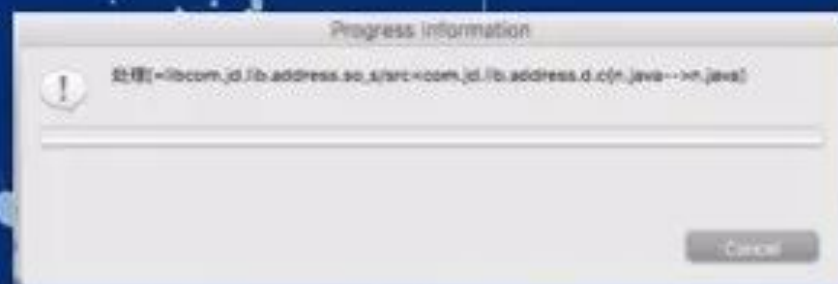
安全+

安全加



# 使用工具搜索

```
projects=all  
pojo=true  
fromFeature=set.*,public,1;;get.*,public,0  
problemFeature=write,invoke,newInstance,forName,create,call,outputStream  
include=*  
level=3
```



# 使用工具搜索

```
89 stream.defaultReadObject()  
90 new IteratorPool(this)  
91 new javax.xml.transform.TransformerException(cnfe)  
92 */  
93 import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;  
94 //method  
95 /*Method:—>readObject(Ljava.io.ObjectInputStream; is);  
96 System.getSecurityManager()  
97 SecuritySupport.getSystemProperty(DESERIALIZE_TRANSLET)  
98 temp.length()  
99 temp.equalsIgnoreCase("true")  
100 new ErrorMsg(ErrorMsg.DESERIALIZE_TRANSLET_ERR)  
101 new UnsupportedOperationException(err.toString())  
102 err.toString()  
103 is.defaultReadObject()  
104 is.readBoolean()  
105 is.readObject()  
106 new TransformerFactoryImpl()  
107 getTransletInstance();  
108 defineTransletClasses()  
109 _class[_transletIndex].newInstance()  
110 translet.postInitialization()  
111 translet.setTemplates(this)  
112 translet.setServicesMechnism(_useServicesMechanism)  
113 translet.setAllowedProtocols(_accessExternalStylesheet)  
114 translet.setAuxiliaryClasses(_auxClasses)  
115 new ErrorMsg(ErrorMsg.TRANSLET_OBJECT_ERR, _name)  
116 new TransformerConfigurationException(err.toString())  
117 err.toString()  
118 */  
119 /*Method:—>getOutputProperties();  
120
```



# 使用工具搜索结果

级别	类	方法	特征
1	com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl	getOutputProperties	None
2	com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl	newTransformer	None
3	com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl	getTransletInstance	newInstance
...			

# 发现的漏洞

- FastJson Remote Code Execution
- 阿里归零实验室2018年4月13日公布了漏洞细节
- Jackson Remote Code Execution
- XStream Remote Code Execution
- AMF Java Remote Code Execution



# 两个未公布的POC关键类→json反序列化

org.jboss.util.loading.ContextClassLoaderSwitcher

```
public void setContextClassLoader(ClassLoader cl)
{
    setContextClassLoader(Thread.currentThread(), cl);
}

public void setContextClassLoader(final Thread thread, final
ClassLoader cl)
{
    AccessController.doPrivileged(new PrivilegedAction()
    {
        public Object run()
        {
            thread.setContextClassLoader(cl);
            return null;
        }
    });
}
```

# 两个未公布的POC关键类→json反序列化

## 调用链

json反序列化ContextClassLoaderSwitcher



ContextClassLoaderSwitcher.setContextClassLoader  
classLoader可利用的比较多，如  
com.sun.org.apache.bcel.internal.util.ClassLoader



Json反序列化类A，如com.sun.org.apache.bcel.internal.util.ClassLoader  
支持的\$\$BCEL\$\$字符串反序列化



# 两个未公布的POC关键类→amf反序列化

org.apache.commons.beanutils.BeanMap

```
public Object put(Object name, Object value) throws IllegalArgumentException, ClassCastException {
    if ( bean != null ) {
        Object oldValue = get( name );
        Method method = getWriteMethod( name );
        if ( method == null ) {
            throw new IllegalArgumentException( "The bean of type: " +
                bean.getClass().getName() + " has no property called: " + name );
        }
        try {
            Object[] arguments = createWriteMethodArguments( method, value );
            method.invoke( bean, arguments );

            Object newValue = get( name );
            firePropertyChange( name, oldValue, newValue );
        }
        catch ( InvocationTargetException e ) {
            logInfo( e );
            throw new IllegalArgumentException( e.getMessage() );
        }
        catch ( IllegalAccessException e ) {
            logInfo( e );
            throw new IllegalArgumentException( e.getMessage() );
        }
        return oldValue;
    }
    return null;
}
```

# 两个未公布的POC关键类→amf反序列化

## 利用链

org.apache.commons.beanutils.BeanMap

bean → org.apache.tomcat.dbcp.dbcp2.BasicDataSource  
\_driverClassLoader → com.sun.org.apache.bcel.internal.util.ClassLoader  
\_driverClassName → \$\$BCEL\$\$\$ xxx payload  
\_logWriter → java.io.PrintWriter

biazeds在javabean的setter方法中过滤了classloader类型  
但是在map的put方法中没有检查  
利用beanMap绕过classloader检查  
利用setLogWriter触发漏洞

```
public void setLogWriter(PrintWriter logWriter)
    throws SQLException
{
    createDataSource().setLogWriter(logWriter); //最终会使用classloader加载类
    this.logWriter = logWriter;
}
```



# 安卓App漏洞分析

## 下载App

- 应用市场
- 自动爬虫
- 更新跟踪

## dumpDex

- mumu
- xposed
- dumpDex  
改进版

## 反编译

- JADX增强

## 导入IDE

- 生成  
project
- 批量处理

## 问题分析

- 攻击面查  
找
- 路径分析

## 漏洞验证

- 参数构造
- 常见命令

# 安卓App漏洞分析

3.0.1 (2017.07.17)

状态 ☐ 未扫描 ☐ 待扫描 ☐ 扫描中 ☒ 扫描结束 应用名

包名

公司名

查询

批量扫描

上一页

1

2

3

4

5

下一页

共 10 页 630 条



京东饭粒  
扫描



京东又城有食  
扫描



网商寻客  
扫描



自考365  
扫描



支付宝  
扫描



必用助手  
扫描



微众银行  
扫描



腾讯自选股  
扫描



王者人生  
扫描



TIM  
扫描



QQ  
扫描



心悦俱乐部  
扫描



手淘宝-语音开屏  
扫描



CMT助手  
扫描



掌上穿越火线  
扫描



掌上英雄联盟  
扫描



QQ国际版  
扫描



王者荣耀助手  
扫描



微信  
扫描



QQ空间  
扫描



绝地求生手游-官方  
扫描



绝地求生官方  
扫描



掌上飞车  
扫描



逆战  
扫描



腾讯地图  
扫描



企鹅汇圈  
扫描



腾讯云叮当  
扫描



企鹅电竞直播  
扫描



QQ浏览器-识一  
扫描



腾讯清理大师  
扫描



微视  
扫描



天天P图  
扫描



布丁相机  
扫描



相册管家  
扫描



DOV  
扫描



微视  
扫描



腾讯时光-照片  
扫描



微同  
扫描



微信读书  
扫描



腾讯动漫  
扫描



腾讯体育  
扫描



腾讯U品  
扫描



QQ邮箱  
扫描



腾讯文档  
扫描



腾讯微云  
扫描



腾讯企点  
扫描



商务微信  
扫描



企业微信  
扫描



企业QQ  
扫描



腾讯新闻  
扫描



海豚智音  
扫描



黄火营地  
扫描



天天快报  
扫描



天天快报大字版  
扫描



天天快报故事版  
扫描



腾讯课堂-在线一  
扫描



ABCMouse  
扫描



企鹅辅导  
扫描



腾讯翻译君  
扫描



腾讯英语君  
扫描



安全加



# 安卓App漏洞分析



下载二维码

文件名

搜索

安装介绍

获取DATA

登录入口

路径 /media/layound/Data/

卸载介绍

执行测试

私有入口

重新扫描

下载介绍

下载代码



测试二维码

代码入口

扫描结果

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

apk

com.kuailest.video.ShareActivity --extra string startPackage str --extra string startAction str --extra string startActivity str --extra boolean resultDataFlag true --extra string gotoActivity str

com.kuailest.video.ShareActivity

com.kuailest.video.MainActivity

com.sina.weibo.sdk.share.WbShareTransActivity

apk

com.kuailest.video.EntryActivity

onCreate(Bundle): V#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

start(Intent): Z#com.kuailest.video.EntryActivity.EntryActivity

```
protected void onCreate(Bundle bundle) {  
    ..... super.onCreate(bundle);  
    ..... Intent intent = getIntent();  
    ..... if (bundle != null) {  
        ..... this.startActivityName = bundle.getString("startActivity");  
        ..... this.flag = bundle.getBoolean("resultDataFlag", false);  
        ..... return;  
    ..... }  
    ..... this.flag = true;  
    ..... this.startActivityName = intent.getStringExtra("startActivity");  
    ..... intent.putExtra("startFlag", -1);  
    ..... Intent intent2 = new Intent("com.sina.weibo.sdk.action.ACTION_WEIBO_AC  
    ..... intent2.putExtras(intent.getExtras());  
    ..... intent2.setPackage(intent.getStringExtra("startPackage"));  
    ..... intent2.setAction(intent.getStringExtra("startAction"));  
    ..... String packageName = getPackageName();  
    ..... intent2.putExtra("_weibo_sdkVersion", "0031405000");  
    ..... intent2.putExtra("_weibo_appPackage", packageName);  
    ..... intent2.putExtra("_weibo_appKey", WbSdk.getAuthInfo().getAppKey());  
    ..... intent2.putExtra("_weibo_Flag", 538116905);  
    ..... intent2.putExtra("_weibo_sign", MD5.hexdigest(Utility.getSign(this, po  
    ..... try {  
        ..... if (TextUtils.isEmpty(intent.getStringExtra("gotoActivity"))) {  
            ..... startActivityForResult(intent2, 705);  
            ..... return;  
        ..... }
```

# 安全

安全加

# 安卓App漏洞分析

执行测试

```
-e startPackage str -e startAction str -e startActivity str --ez resultDataFlag true -e gotoActivity str
```

执行

逐步

命令 返回结果

```
adb -s 2c3d3d62 shell am start -n  
com.xiaomi.apps.videodaily/com.sina.weibo.sdk.share.WbShareTransActivity -e startPackage str -  
e startAction str -e startActivity str --ez resultDataFlag true -e gotoActivity str
```

结果

```
Starting: Intent {  
cmp=com.xiaomi.apps.videodaily/com.sina.weibo.sdk.share.WbShareTransActivity (has extras) }
```

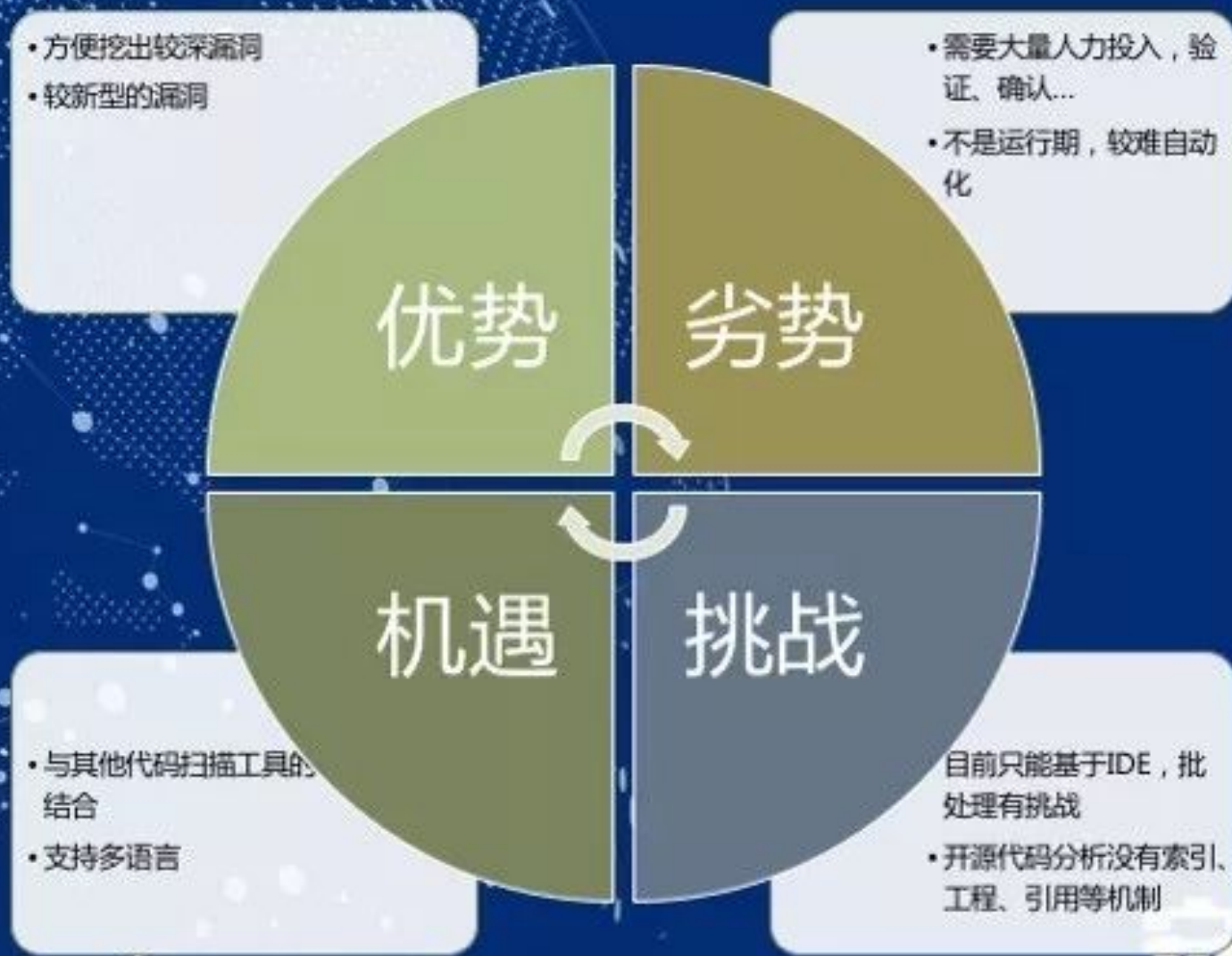
关闭



# 发现的App漏洞

- 淘宝、天猫
- 支付宝、蚂蚁财富
- 京东
- ...

# 基于源代码漏洞分析方法的SWOT分析





# 谢谢

