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 WiFiMasterKey Security Response Center

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

协办方

Alibaba Security Response Center | 饿了么安全应急响应中心

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

安全猎携 SECPULSE

iQIY

唯品会安全应急响应中心

美丽联合安全

安全加

基于源代码的漏洞挖掘

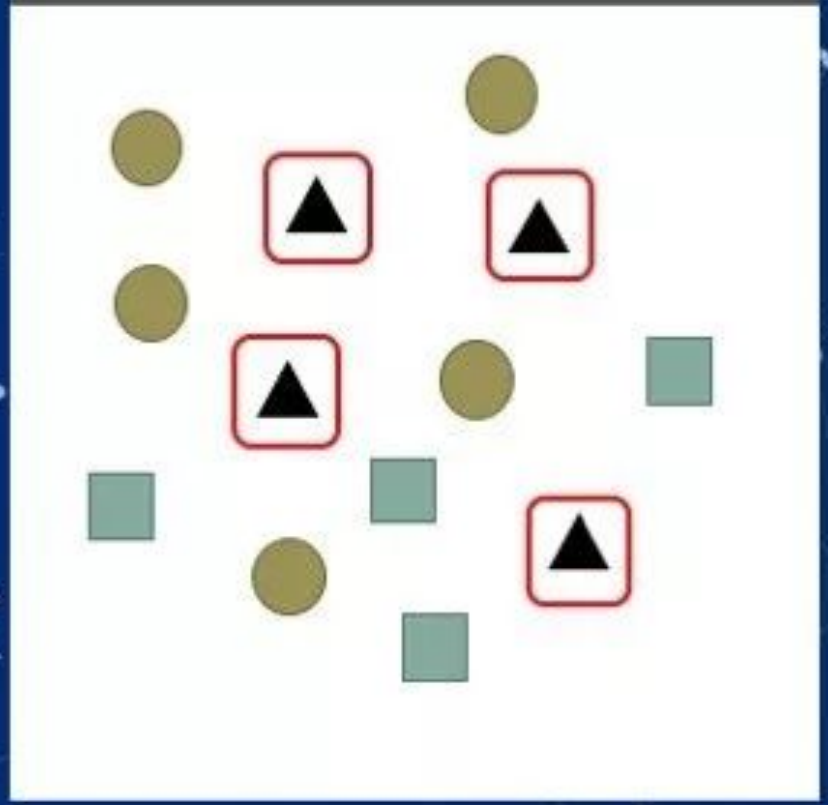
JAVA

- 漏洞分析思路和工具
- 未公开的两个POC
- 安卓App漏洞分析

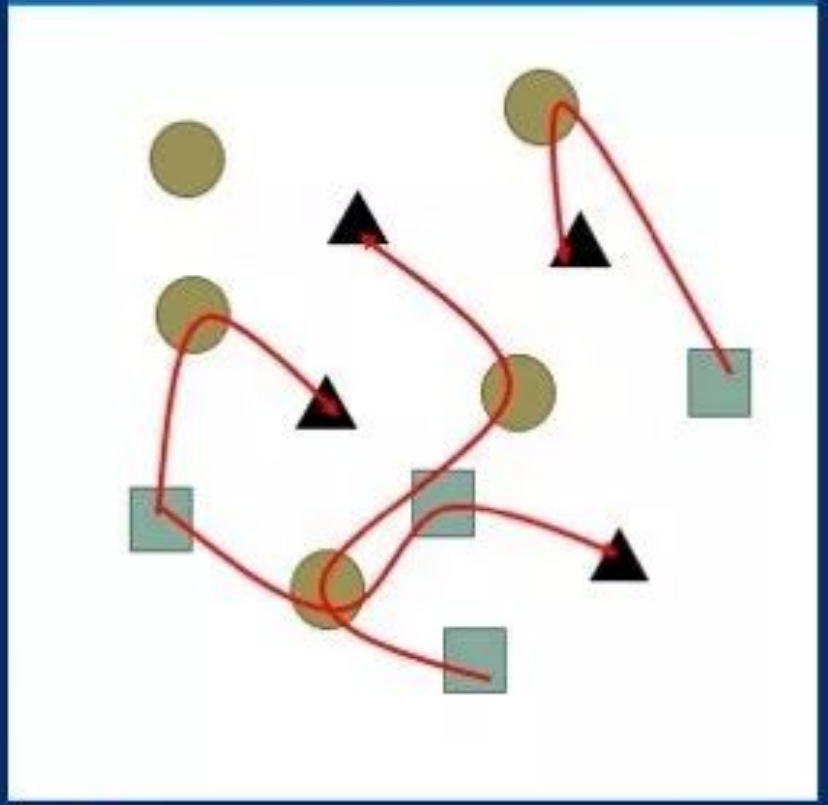
安全+
安全加

核心思路→基于代码路径的分析

基于文本和规则的扫描



基于代码路径的分析



攻击面　问题代码　正常代码

安全+
安全加

基于路径分析的核心

基于IDE	索引	攻击面识别	搜索
• 代码 • 环境 • 依赖	• callee • caller • extend • implement	• 反序列化 • 安卓	• 正向搜索 • 反向搜索

安全+
安全加

为什么基于IDE？

安全加

为什么基于IDE？

安全加

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

class1

Class1是pojo但是没有漏洞

extends

class2

Class1不是pojo但是有漏洞

extends

class3

Class1是pojo但是没有漏洞

安全加

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
    ...
}
```

安全加

为什么需要继承或接口实现索引？

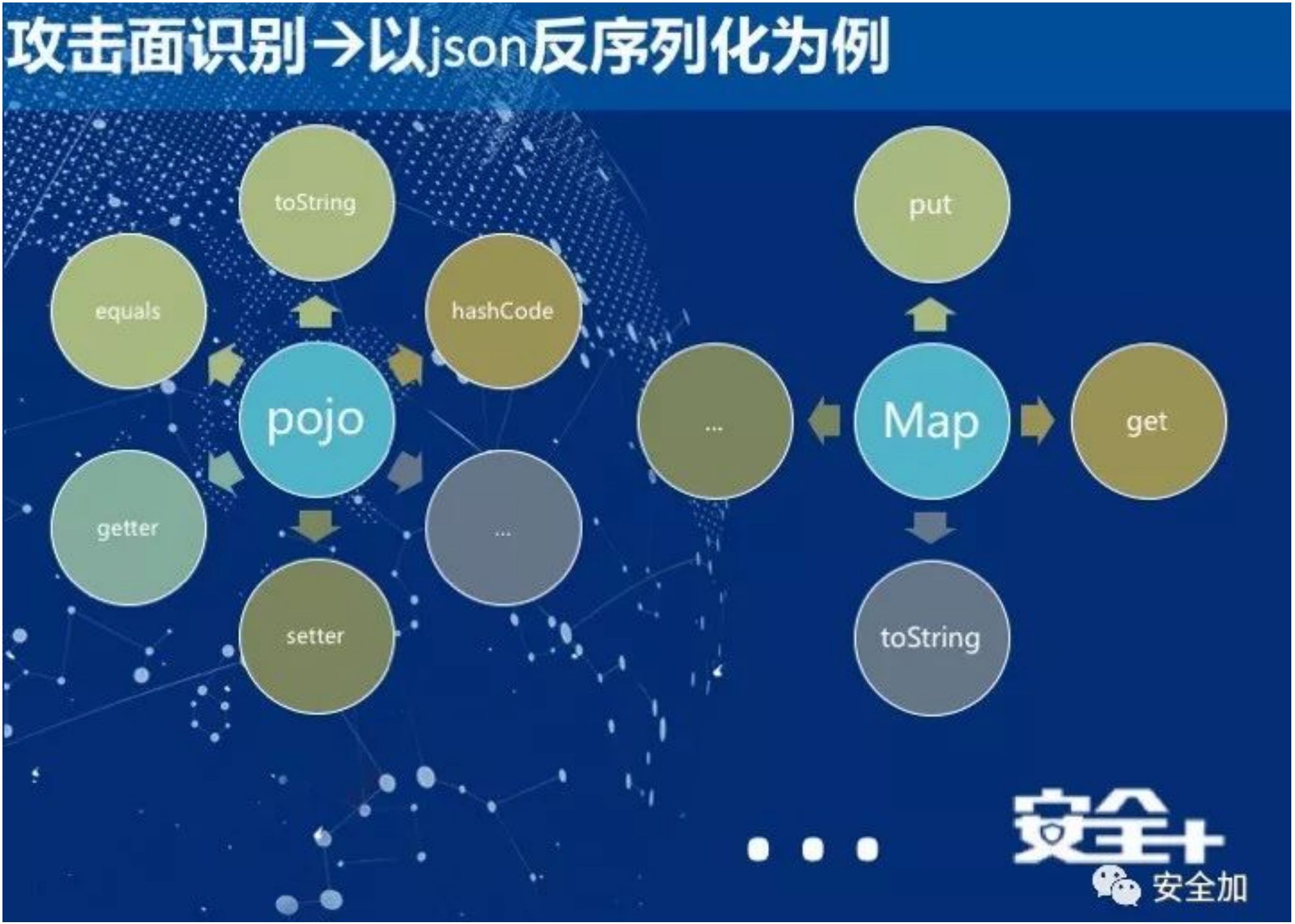
```
Intent intent = new Intent(Intent.ACTION_VIEW,
TestActivity.class);
startActivity(intent);
```

TestActivity的方法

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

安全加

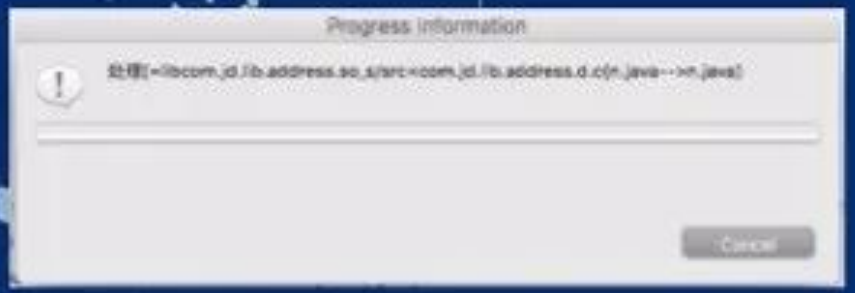
攻击面识别→以json反序列化为例



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
```



安全加

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();
```

安全加

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
...			

安全加

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

安全+
安全加

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;
      }
    });
  }
```

安全加

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

安全加

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;
    }
```

安全加

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;
}
```

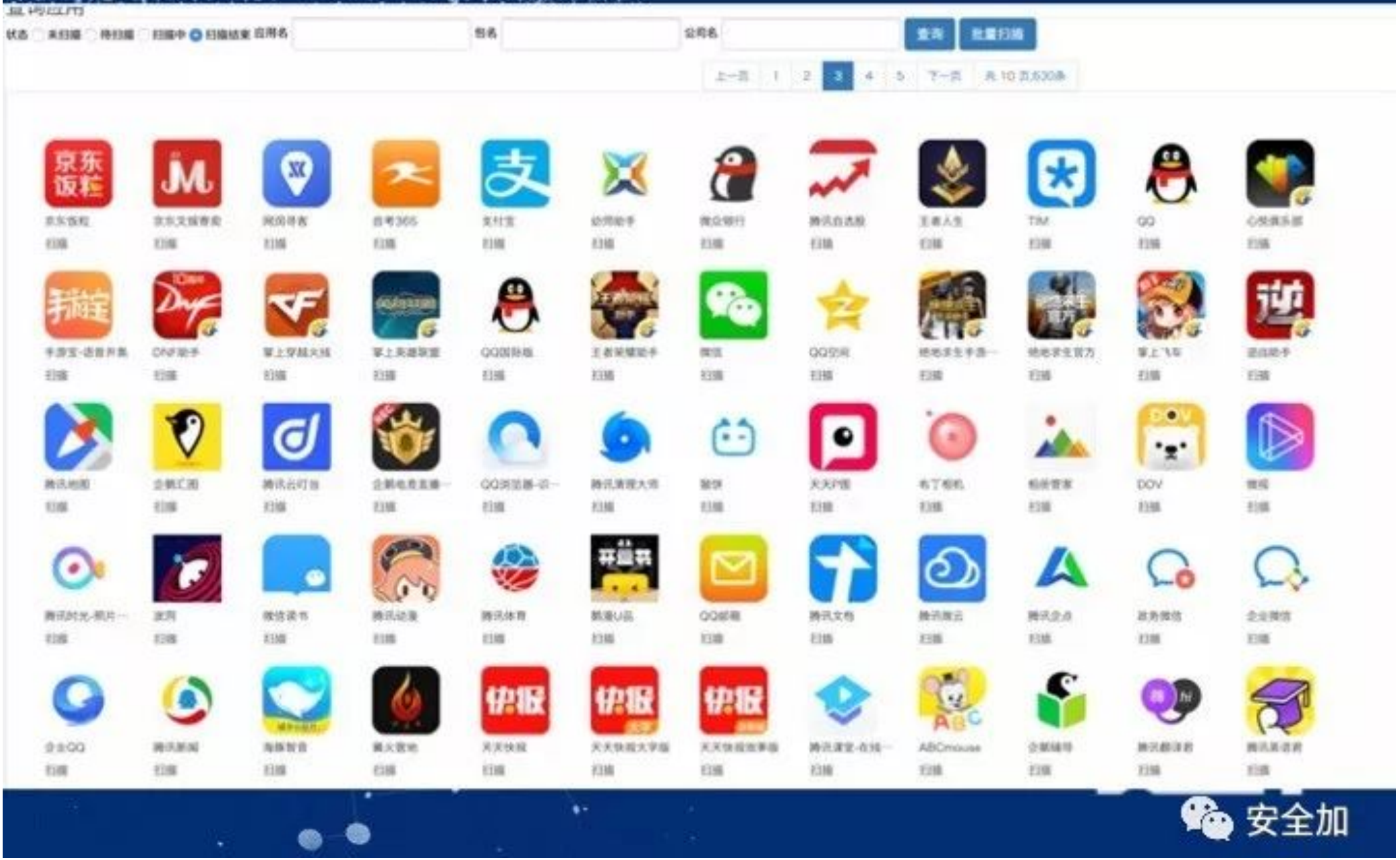
安全加

安卓App漏洞分析

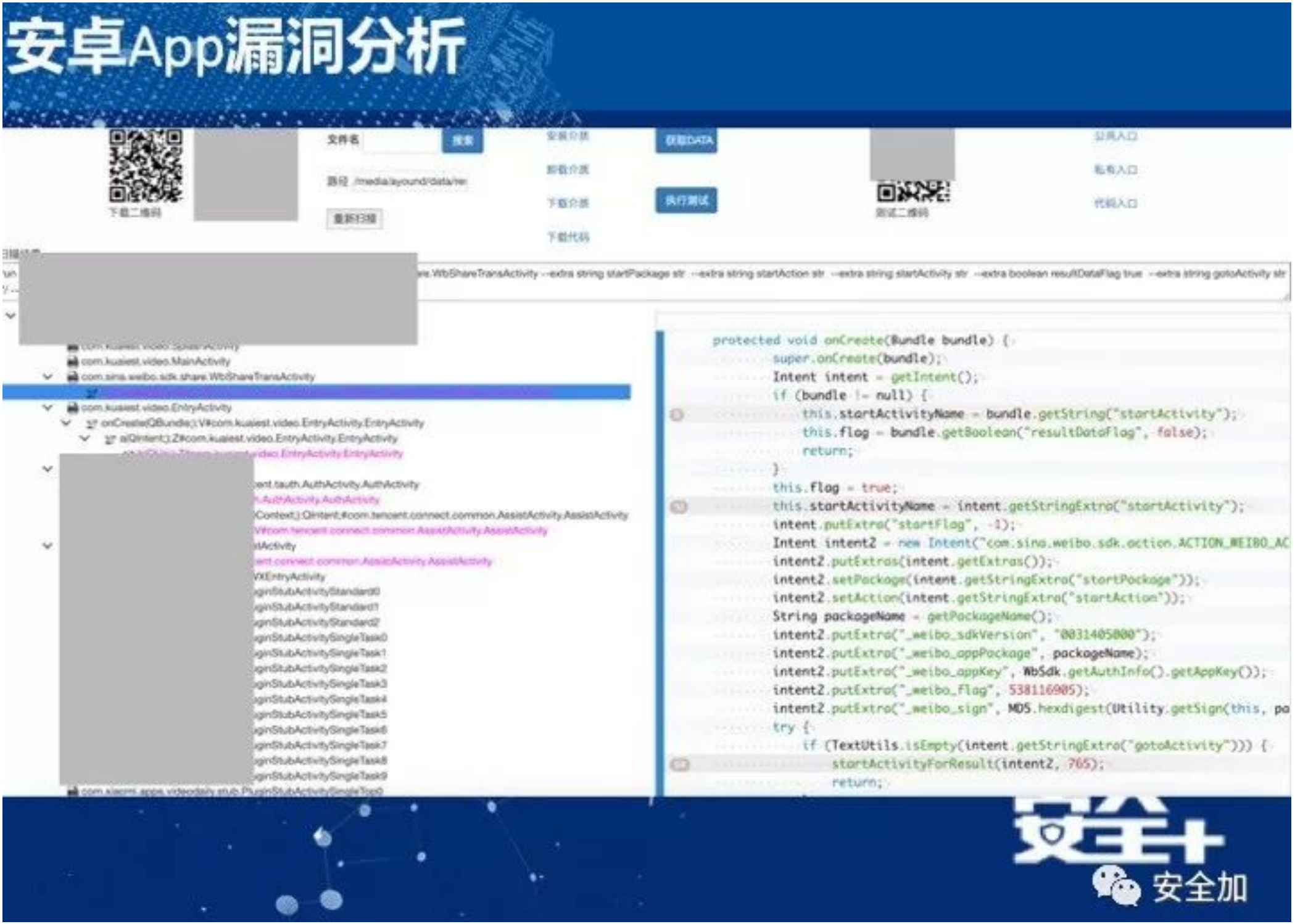
下载App	dumpDex	反编译	导入IDE	问题分析	漏洞验证
• 应用市场 • 自动爬虫 • 更新跟踪	• mumu • xposed • dumpDex 改进版	• JADX增强	• 生成 project • 批量处理	• 攻击面查找 • 路径分析	• 参数构造 • 常见命令

安全+
安全加

安卓App漏洞分析

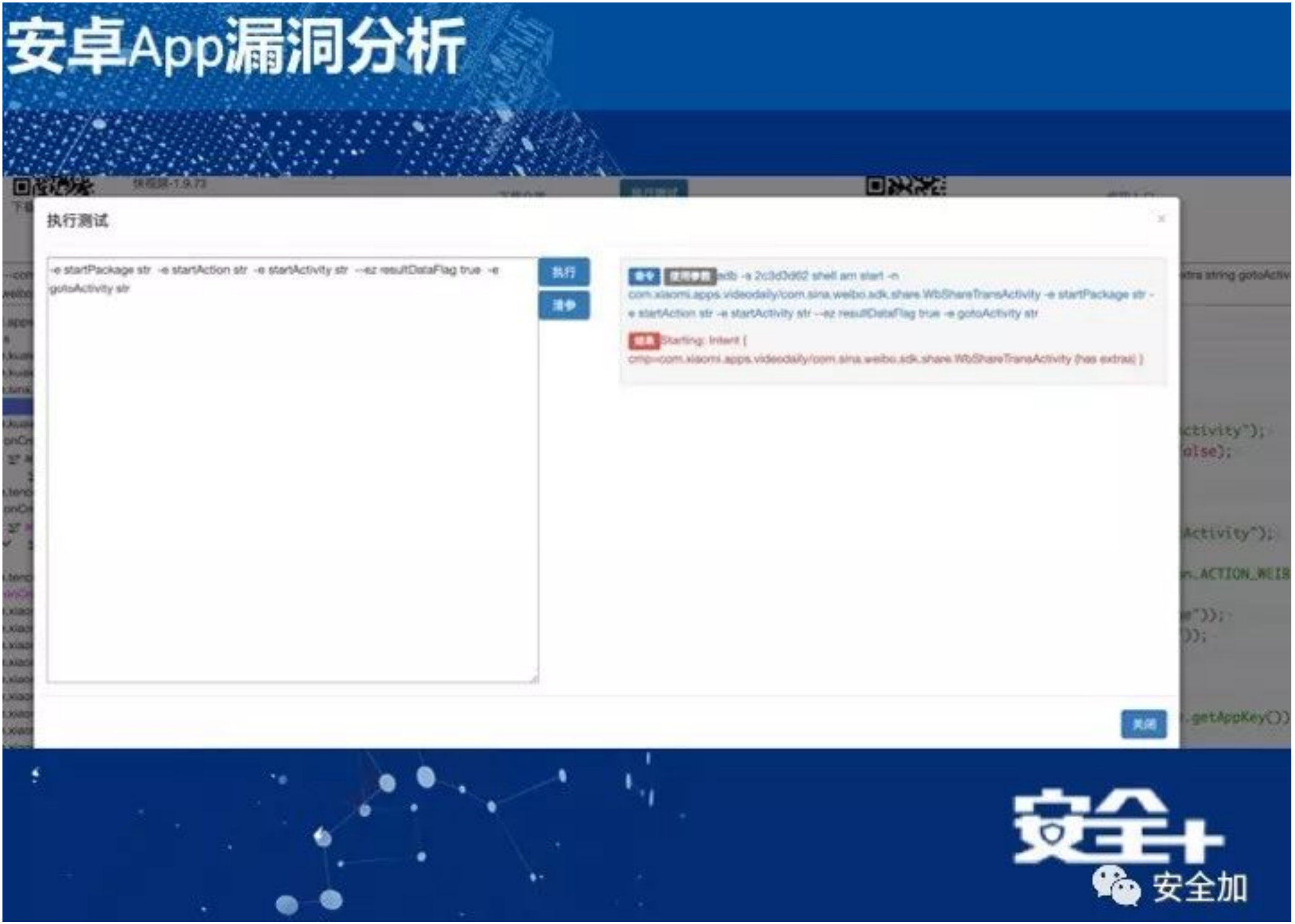

状态
应用名
包名
公司名
查询
批量扫描
京东饭粒
支付宝
QQ
微信
王者荣耀
DNF助手
腾讯地图
腾讯视频
腾讯体育
腾讯文档
腾讯微云
腾讯企点
企业微信
腾讯新闻
ABCmouse
安全加

安卓App漏洞分析



安全加

安卓App漏洞分析



安全加

发现的App漏洞

- 淘宝、天猫
- 支付宝、蚂蚁财富
- 京东
- ...

安全+
安全加

基于源代码漏洞分析方法的SWOT分析

安全加


谢谢
安全+
安全加