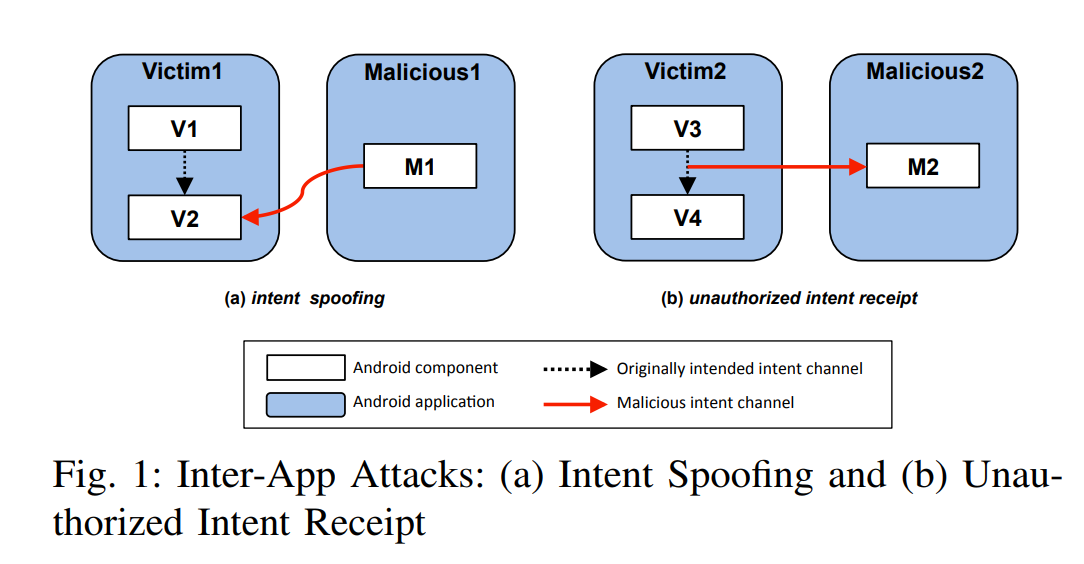
**SEALANT: A Detection and Visualization Tool for Inter-app Security Vulnerabilities in Android**

两个例子激发了他们这个工具的开发。

1、



恶意程序malicious1 的组件M1可以向受害者应用程序victim1的组件M2发送intent，用来利用victim1的功能。比如，V2被设计为从V1接收intent中的手机号码并向这个号码发送信息，M1向V2发送了伪造的intent和不安全的号码，那V2就会把信息发送给非预期的号码。这种情况下，M1和V2之间就存在脆弱的ICC路径。

Intent被发送或没有被特别保护的广播中时，任何组件都可以通过声明符合intent的特征来接收它。比如V3广播一个intent来将账户信息传递给V4，M2可以窃听intent中的敏感信息。这种情况下，M2和V3之间存在脆弱的ICC路径。

1、**SEALANT客户端**使用户可以跟SEALANT后端进行交互，用户选择APK文件，客户端就将文件转发给SEALATN Core，客户端会将后端返回的组合信息呈现为可剪裁的视觉符号。

可选择联系用户以评估每个易受攻击的ICC路径。参考专家的评估可以帮助非专家用户避免错误识别的路径。

2、**SEALANT Core** 负责控制每个操作，并将分析的信息转换为可视化表示的格式。当请求一组APK文件时，Core组件首先检查之前的分析结果是否存储在SEALANT Repository中，有的话会重用它们。否则反编译APK文件并传递给SEALANT Analyzer。Core随后将从Analyzer中收到的应用程序模型和易受攻击的ICC路径合并转换为与Client兼容的预定义格式。Core将来自Client的用户评估存储在Repository中。

3、SEALANT Analyzer由两个组件组成：模型提取模块和漏洞标识模块

**模型提取模块**检查每个应用的Android manifest文件和字节码以提取程序的架构信息（组件，intent filter，权限和intent），对于每个组件，它都在敏感的API和ICC-call API之间执行数据流分析。包含这种数据流的组件被标记为易受攻击的组件。然后建立一个捕获到所有程序信息的总结模型并存储在Repository中。

**漏洞标识模块**首先根据Android API参考文档中的规则，通过匹配每个组件的intent和intent filter来构建一个ICC图。ICC图中，一个节点表示一个组件，一个边表示为元组<s, r, i>,s是intent的发送组件，r是intent的接受组件，I 是他们之间的intent。

遍历这个ICC图，根据两点来标记出这个边是否是易受攻击路径：

1. 两端是否都是易受攻击组件
2. 是否是特定组合ICC模式的一部分

具体而言：

两个不同的边（一个跨应用，一个在程序内）同时指向同一个组件，而这两个边的i却是相同类型的intent，这就表示这个ICC路径易受到intent欺骗攻击。

两个不同的边（一个跨应用，一个在程序内）同时从一个组件出发，而这两个边的i却是相同类型的intent，表示这个ICC路径易受到未经授权的intent接收攻击。识别出的ICC路径存储在Repository中。

1. **SEALANT Repository**

模型数据库：

SEALANT Repository通过包的名称和版本号管理每个应用程序的模型，

如果用户设备上安装或者更新应用程序时，SEALANT仅从安装或更新的应用程序中提取架构信息，重复使用先前的分析结果。

路径数据库：

维护已经识别的易受攻击的ICC路径，还会维护每个用户的评估。