Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps

镜像：通过移动伴侣应用程序评估物联网设备安全性

28th USENIX Security Symposium 2019

目录

[Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps 1](#_Toc38117291)

[镜像：通过移动伴侣应用程序评估物联网设备安全性 1](#_Toc38117292)

[28th USENIX Security Symposium 2019 1](#_Toc38117293)

[挑战 2](#_Toc38117294)

[摘要 2](#_Toc38117295)

[背景 2](#_Toc38117296)

[相关工作 2](#_Toc38117297)

[主要贡献 3](#_Toc38117298)

[概念 3](#_Toc38117299)

[主要工作 4](#_Toc38117300)

[1.系统设计 5](#_Toc38117301)

[1.1 IOT APP收集器 5](#_Toc38117302)

[1.2 APP 分析引擎 5](#_Toc38117303)

[1.2.1 设备接口分析 6](#_Toc38117304)

[1.2.2 提取响应 6](#_Toc38117305)

[1.2.3 成对的请求和响应 6](#_Toc38117306)

[1.2.4 请求字符串重建 7](#_Toc38117307)

[1.3 设备印记分析 7](#_Toc38117308)

[1.4 fuzzing hash 分析 7](#_Toc38117309)

[1.5 模块化问题 8](#_Toc38117310)

[1.6 跨APP 分析引擎问题 8](#_Toc38117311)

[结果及注意的问题 9](#_Toc38117312)

[1问题 9](#_Toc38117313)

[2 准确性问题 10](#_Toc38117314)

[局限性和未来工作 11](#_Toc38117315)

[结论 11](#_Toc38117316)

[个人心得 12](#_Toc38117317)

# 挑战

针对于智能家居物联网设备的安全性问题，我们为了识别这些易受攻击的设备，需要对真实设备或其固件映像分析，但以上两者都不容易获取

# 摘要

本文提出一种平台，来加速易受攻击设备的发现和分析。

这种平台的实现方法**基于两个前提**：

（1）IOT设备组件重用率高，因此在一台设备中发现的漏洞通常在其他设备中存在

（2）借助伴随APP来间接推断重用组件

通过对移动伴侣应用的交叉分析，我们可以估算设备之间的相似度。

我们的研究揭示了智能家居IoT设备之间易受攻击组件的共享

# 背景

1. 攻击面的增加，传统攻击的融合
2. 相当一部分中小型企业的设备缺乏最佳实践的控制
3. 设备相对便宜，不能支持昂贵的安全基础设施
4. 供应商碎片化的分发漏洞等

**解决方案：在部署之前识别出易受攻击的设备，并采取适当的措施，**升级设备固件、识别和阻止可能利用该漏洞的流量，或完全隔离该设备

# 相关工作

1. 针对于互联网规模的扫描，以检测公共可访问的普通易受攻击的设备
2. 集中在静态或者动态分析物联网设备或固件的分析，但存在如上文提到的不容易获取的问题

# 主要贡献

* 展示了物联网设备配套的移动应用如何为设备本身的安全性评估提供帮助。
* 该方法在物理设备和固件映像不可用的情况下评估物联网设备的安全状态，证明了该方法的有效性。
* 提出了一个大规模收集、过滤、分析和聚类移动应用的平台
* 报告称，来自73个不同供应商的324个设备可能会引起安全问题。

# 概念

* **符号执行：**在计算机科学中，符号执行技术指的是通过程序分析的方法，确定哪些输入向量会对应导致程序的执行结果向量的方法。通俗的说，如果把一个程序比作DOTA英雄，英雄的最终属性值为程序的输出（包括攻击力、防御力、血槽、蓝槽），英雄的武器出装为程序的输入（出A杖还是BKB）。那么符号执行技术的任务就是，给定了一个英雄的最终属性值，分析出该英雄可以通过哪些出装方式达到这种最终属性值效果。
* **点对点连接（P2P）:** 又称对等互联网络技术，是一种网络新技术，依赖网络中参与者的[计算能力](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E8%83%BD%E5%8A%9B/4554810)和带宽，而不是把依赖都聚集在较少的几台服务器上。P2P网络通常用于通过[Ad Hoc](https://baike.baidu.com/item/Ad%20Hoc)连接来连接[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9/865052)。这类网络可以用于多种用途，各种文件共享软件已经得到了广泛的使用。[P2P技术](https://baike.baidu.com/item/P2P%E6%8A%80%E6%9C%AF)也被使用在类似[VoIP](https://baike.baidu.com/item/VoIP/110300)等实时媒体业务的[数据通信](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%80%9A%E4%BF%A1/897073)中。
* **程序切片：**程序切片技术是一种重要的程序分析技术，广泛应用于程序的调试、测试与维护等领域。程序切片主要通过寻找程序内部的相关特性，从而分解程序，然后对分解所得的程序切片进行分析研究，以此达到对整个程序理解和认识的目的。而动态程序切片主要是指在某个给定输入的条件下，源程序执行路径上所有对程序某条语句或兴趣点上的变量有影响的语句。解决在程序理解、程序复杂性度量、程序转换和评测中遇到的问题。基于数据流图或依赖图实现
* **Fuzzing:** 通过构造测试输入，对软件进行大量测试来发现软件中的漏洞的一种模糊测试方法
* **SSdeep:** 这是一个基于模糊哈希（Fuzzy Hashing）算法的工具
* **Fuzzing Hash:** 又叫基于内容分割的分片分片哈希算法，主要用于文件的相似性比较。主要被用来取证，恶意代码检测，以及开源软件漏洞挖掘。
  + **原理：**使用一个**弱哈希**计算文件局部内容，在特定条件下对文件进行**分片**，然后使用一个**强哈希**对文件每片计算哈希值，取这些值的一部分并连接起来，与分片条件一起构成一个模糊哈希结果
  + 通俗来说，对**文件的部分变化**，比如多处修改、增加、删除部分内容，**只会改变模糊哈希结果某一个分片的值**，并不会影响整体的变化，使用模糊哈希均能发现与源文件的相似关系，也就是**判断相似度**的一种方法。这种方法对于**源码级样本**的比对，比如*webshell*，是非常有效的。

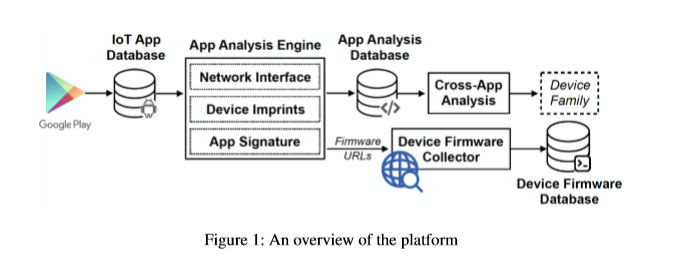
# 主要工作

我们的平台，我们试图加快识别脆弱设备的过程，提供两个功能：

**应用分析**：通过分析其伴侣应用找到一个设备的特点

**跨应用分析：**发现设备所属的族，在应用分析中通过分析多个应用程序发现了一些相似性特征，从而有助于识别基于共享组件的具有类似漏洞的应用程序。

## 1.系统设计



### 1.1 IOT APP收集器

**预处理：**爬了谷歌Play Store2。我们总共下载了3094个Android应用程序，其中2081个包含在最终数据集中

**挑战：**识别哪些应用程序是物联网设备的移动配套应用程序

**解决方案：**从智能家居产品数据库SmartHomeDB[5]中手动选择281个种子应用初始化爬虫，并通过谷歌Play上种子应用与其他应用之间的连接(例如，关键字、建议和类别)，使用滚雪球采样来收集更多的应用

**问题：**收集的存在噪声/过滤

**过滤方法：**过滤基于一个集群模型(亲和传播[26])，该模型根据应用程序在安装时请求的权限和应用程序在运行时可能调用的敏感Android api来对应用程序进行集群划分

### 1.2 APP 分析引擎

应用程序分析引擎分析收集到的移动伴侣应用程序，以便为单个设备构建设备配置文件

**目标:** 间接地从应用程序计算设备的样子

**三种方法：**

* 设备接口分析计算的网络接口设备
* 独特的印记分析计算字符串可能有关的设备
* 一个模糊散列分析计算代码签名的移动伴侣应用

### 1.2.1 设备接口分析

**原理**：设备接口通常很好地反映了设备的样子.（设备所使用的协议、设备运行的服务、设备支持的功能，有时还包括设备所使用的硬件组件）

应用程序和设备之间的点对点连接有助于这种估计。

中小型供应商生产的大部分设备(76.3%)支持应用程序和设备之间的点对点连接。甚至像TPLink这样的大型供应商也支持云模式和点对点模式，以应对网络故障和隐私问题。可以实现设备接口的一体化

**方法：**我们使用逆向方法来计算应用程序的网络接口，从应用程序可能接收到的网络响应消息开始，因为这些消息是设备输出的信息。我们首先确定应用程序中的消息处理函数，然后静态地确定响应消息的外观

符号约束的连接被存储为响应消息的“描述”

### 1.2.2 提取响应

**方法：**依赖于符号执行（前向推导）[33]来估计来自设备的响应消息的样子，而不需要实际运行设备

**目的：**提取响应形式

首先建立了一个伴随应用的控制依赖图*Control Dependency Graph (CDG)*和 数据依赖图*Data Dependency Graph (DDG)*通过Soot。

**触发**：通过触发应用程序的改变来决定是否终止符号执行，有效响应往往伴随着装填更改

### 1.2.3 成对的请求和响应

**目的：**识别触发设备响应程序的请求

匹配的请求和响应常常共享它们处理功能的一个大型代码库(即，用于处理请求和响应的类和方法)。这种相似性反映在运行时的堆栈中。请求和响应是基于公共堆栈帧(例如，由同一用户单击触发的堆栈帧)进行匹配的，尽管它们位于运行在不同线程中的不同类中。

### 1.2.4 请求字符串重建

**目的：**重构请求字符串，请求是由应用程序生成的

**方法：**通过程序切片和执行来重建程序值的技术[12,29,48,56]

**输出：**设备接口分析产生的结果是一组请求和响应对。请求是完全或部分重构的请求字符串，响应是一组符号约束。

**总结：**设备接口分析产生的结果是一组请求和响应对。请求是完全或部分重构的请求字符串，响应是一组符号约束。一个设备据说有一个相似的接口，另一个设备可能同时接受相似的请求和输出相似的响应。

### 1.3 设备印记分析

**设备痕迹：**在应用程序中找到的唯一字符串可以帮助关联不同的设备。我们对应用程序和设备之间的通信中出现的印迹特别感兴趣，因为它们代表了设备的独特性。

**提取痕迹重点：**设备（后门）关键字，证书和密钥，非平凡的用户名和密码以及特殊URL。

**方法：**构建应用程序的数据依赖图，然后从网络API向后检查，以找到影响这些API参数的应用程序中的常量字符串，解析器会忽略用户名和密码比如admin

**问题：**尽管印记可以作为相关性的有力证据，但由于很多时候设备的印记不会在应用程序中显示出来，因此作为一种方法的印记分析通常不太适用。例如用于调试作用的魔术关键字，而且设备调试不是面向客户的关键功能

**因此我们不推荐此方法**

### 1.4 fuzzing hash 分析

**Fuzzing:** 通过构造测试输入，对软件进行大量测试来发现软件中的漏洞的一种模糊测试方法

**Fuzzing hash 及 ssdeep 见前文概念部分**

包括类、库和其他类型的资源(例如文本)，并比较应用程序之间的结果。

使用模糊哈希的好处是我们可以将相似但不完全相同的对象关联起来。

不过，请注意，设备之间的相似性并不一定意味着应用程序之间的相似性。我们注意到，在许多情况下，相似的设备有不同的应用程序。，应用程序是独立开发的)，因此导致模糊哈希分析失败。

### 1.5 模块化问题

**原因：**有IOT设备模块化相似性的问题，IoT设备供应商（尤其是规模较小的设备供应商）通常由市场上许多现有模块组成其产品。

设备组件经常被应用程序中不同的代码模块**管理**(例如，类，包)

我们的分析也进行模块化也很重要，以便以单个设备组件的更细粒度跟踪设备相似性并检测漏洞传播

因此，我们从应用程序中推断出这种模块化(例如，基于类层次结构和调用堆栈)，并将上述分析方法应用于各个模块。

### 1.6 跨APP 分析引擎问题

**目标：**检测不同设备之间的模块相似性

**相似性：**相似的软件组件、相似的硬件组件、相似的协议和相似的后端服务。

**相似的软件组件：**类似设备的接口，特别是应用程序接口，表明了不同设备的软件组件之间的强大连接。这种关联是强大的，因为在很多时候，安全弱点表现在软件和安全的弱点上，一个设备的安全性能会直接影响到其他设备的安全性能。

**重新品牌化：**规模较小的供应商有时并不开发自己的产品。相反，他们从原始设备制造商定制物联网设备，然后用自己的品牌转售

**相似的硬件组件：**反映在设备配套应用程序中，因为应用程序需要配置硬件组件或与硬件组件进行交互。由于硬件的特殊性，这样的设备-应用程序接口可以是唯一的，允许使用相同硬件的不同设备之间有很强的相关性。

**类似的协议和后端服务**

* 特定的协议通常有自己的请求和响应格式
* 特定的后端服务常常公开标准的api
* 跨应用分析引擎可以检测网络接口的相似性，从而将使用相同协议或使用相同后端服务的设备关联起来，即使这些协议或后端服务没有文档记录。

**未来工作 物联网设备的安全评估可以从相似性评估中获得受益**

* 相同的开发人员或分包商可能采用类似的编码方式，从而在他们的设备中构建了相同的一组糟糕的安全实践或漏洞
* 相同的开发工具链(例如，编译器)可能以类似的方式转换代码，从而导致相同的安全问题集
* 计划将我们的分析扩展到更多的相似性维度，以获得更准确和完整的智能家居物联网设备评估。

**设备固件收集器**

**目的：**收集固件镜像帮助跨应用分析

**方法：**我们利用固件下载链接内嵌在移动伴侣应用程序

因为物联网设备通常是无头的。他们经常通过配套的应用程序部署固件更新。

注意，不是所有的设备固件都可以下载

# 结果及注意的问题

### 1问题

* 应用程序的混淆

绝大多数(85.8%)的设备配套应用是由Android SDK中的标准工具(即Android SDK)生成的。它通过重命名类、方法和字段来混乱应用程序。

* 应用程序打包

一些开发者使用打包器来加密他们的代码，这也会对网络接口分析产生影响。然而，与之前的研究[53]的观察结果相一致，恶意软件中经常会出现包装程序，而良性应用程序很少使用。

### 2 准确性问题

**本文提出的方法的缺陷是：**这种分析的输出(例如，是一种推测，指出了潜在的安全问题，需要用真实的设备进行验证。

**解决方案：**

1. 尝试获取真实的设备并在本地环境中测试它
2. 模拟或在某些情况下静态分析存储在设备固件数据库中的设备固件
3. 搜索在线报告，包括供应商手册和网站、漏洞报告论坛、物联网黑客社区
4. 与供应商合作，请求他们帮助验证结果

**问题：**错误的讲一个设备标记为易受攻击

**原因：**由于设备模型和易受攻击的设备具有非常相似的接口，然而在真实的设备和固件上的模拟验证，该设备并不脆弱，检查固件发现和易受攻击的设备非常相似，但是易受攻击的代码被删除了，并且发布了补丁。

**假设的原因：**由于这些设备相对较新的市场(即。，在漏洞被报告后)，*KGUARD*可能已经定制了软件配置，并在产品投放市场前修补了漏洞。

**观察到的问题：**尽管这些漏洞已经存在很久了，但我们仍然看到大量设备可能存在漏洞

**可能的原因：**市场高度分散，许多较小的供应商在出售设备后从不费心照顾设备。

**设备重新品牌化问题：**由于市场分散，较小的物联网供应商有时不开发自己的产品。相反，他们从原始设备制造商定制物联网设备，然后用自己的品牌转售。这使产品的安全管理变得复杂，并将客户置于危险之中，因为上游供应商的漏洞往往会传播到更广泛的下游供应商，但安全补丁不会。

**通用的硬件漏洞：**物联网设备供应商面临的挑战是，他们通常不会公布正在使用的硬件组件。有时供应商修改了硬件配置，但是没有清理程序代码，平台仍然会检测出问题

**令人困惑的信任模式：**物联网设备开发人员有时会对设备所处的本地环境产生一种令人困惑(如果不是相互冲突的话)的信任假设

* + 一方面，他们似乎认为当地的环境(即，消费者的家庭网络)是不值得信赖的。他们使用加密和认证来保护应用程序和设备之间的通信。、
  + 另一方面，他们对应用程序的信任程度过高/例如使用硬编码的方法
  + 加密通信使用的加密密钥(即最重要的是，它内置在应用程序中。因此，任何能够访问该应用程序的人都可以轻松地伪造通信。

# 局限性和未来工作

* 从本质上局限于我们可以从应用程序获得的信息，有时我们可以获得的信息可能不是设备的准确反映（一个设备可能已经修补了一个漏洞，而这个补丁根本没有改变设备接口。在这种情况下，我们的分析仍然会将设备输出为潜在的漏洞，因为我们的平台只检查应用程序，不会知道补丁的存在。）
* 网络接口分析可能会降低目前的有效水平，比如物联网后端服务器或云将设备接口与应用程序接口显著分离（一个例子是谷歌和Amazon设备，其中大部分管理是通过云完成的。在这种情况下，我们的方法可以收集较少的设备软件信息。）
* 相似性分析的维度和粒度

对App分析引擎的进一步改进可能允许平台检测设备软件栈的更精细组件(如web服务器、PHP解释器、web应用程序、OS、驱动程序)和其他维度(如web服务器、PHP解释器、web应用程序、OS、驱动程序)的相似性。，类似的开发人员，类似的开发工具链)。这将使我们能够更全面和准确地跟踪漏洞传播。

# 结论

在这篇论文中，我们提出了一个平台来加速在智能家居物联网设备市场中的脆弱设备发现。与以往研究物联网设备或固件镜像的方法不同，我们的平台通过分析设备的移动伴侣应用来间接检测设备相似性和跨设备传播的漏洞，从而为大规模分析提供了可行性。通过分析2081个移动伴侣应用程序，我们的平台能够发现73家供应商的324台设备存在潜在的安全隐患，其中38家供应商的164台设备被证实确实存在安全隐患。

# 个人心得

本文对于安全威胁的发现角度相当的新颖，通过分析相关性的工作同样也可以发表文章。例如针对于漏洞报告的差异性分析，针对差异性的分析报告也是可以成为成果的，是一个好的切入点