Burglars’ IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds

窃贼的物联网天堂:理解并降低物联网云上通用信息协议的安全风险

目录

[Burglars’ IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds 1](#_Toc36406078)

[窃贼的物联网天堂:理解并降低物联网云上通用信息协议的安全风险 1](#_Toc36406079)

[摘要： 1](#_Toc36406080)

[背景： 2](#_Toc36406081)

[亟待解决的问题： 2](#_Toc36406082)

[问题的意义： 2](#_Toc36406083)

[主要贡献： 2](#_Toc36406084)

[基本概念 3](#_Toc36406085)

[应用场景 4](#_Toc36406086)

[主要工作 4](#_Toc36406087)

[1.未经授权的will消息 4](#_Toc36406088)

[2.通过will消息发动攻击 5](#_Toc36406089)

[3.will消息造成的原因 5](#_Toc36406090)

[4.未经授权的retain 消息的攻击 5](#_Toc36406091)

[5.MQTT的会话错误 6](#_Toc36406092)

[6.没有更新的会话生命周期状态 6](#_Toc36406093)

[7.未经身份验证的MQTT身份—没有身份验证 7](#_Toc36406094)

[8.MQTT topic 7](#_Toc36406095)

[9.实验 8](#_Toc36406096)

[10.泄露的MQTT 所引发的问题 8](#_Toc36406097)

[可拓展点 8](#_Toc36406098)

[总结 9](#_Toc36406099)

[心得 9](#_Toc36406100)

## 摘要：

服务于设备-用户通信的（general messaging protocol）有存在不清楚的风险，本文的工作旨在针对于主要的物联网云研究主流的消息传递协议MQTT的安全漏洞，同时本文对其漏洞进行攻击并进一步设计出访问模型MOUCON,并验证该模型.

## 背景：

基于云的物联网系统得到广泛的部署使用，这些云服务的核心是在物联网和用户之间协调通信，然而由于设备的算力限制，设备均会选择用相对简单的MQTT协议来实现需求，然而MQTT起初不是以在不安全环境中设计的，几乎没有内置的认证和授权机制，基于此引出了许多问题诸如丧失设备控制、敏感信息公开等

## 亟待解决的问题：

MQTT起初不是以在不安全环境中设计的，几乎没有内置的认证和授权机制由此所带来的重大安全隐患

## 问题的意义：

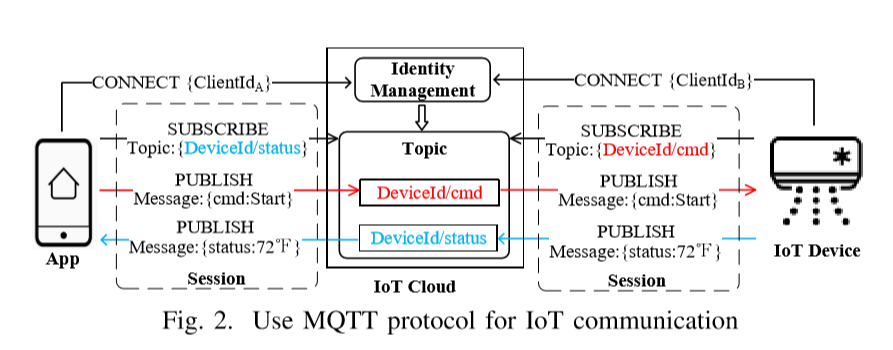
* 帮助主流的云服务厂商规避由MQTT带来的安全风险
* 促进MQTT协议的进一步标准化
* 揭示协议在不安全环境下所带来的安全隐患
* 帮助提出有效的安全设计原则和方法

## 主要贡献：

* 对安全物联网通信有了新的理解 揭示了在简单和良好环境下运行的协议同在复杂环境下运行之间的差距。
* 使用适当的安全手段：通过扩展协议来覆盖新的场景，撤销和保护现有的安全敏感状态和转换
* MQTT服务器面临的安全风险的普遍性揭示
* 提出新的安全设计原则和实现方法

## 基本概念

* **会话劫持：**分为中间人和注射式攻击。嗅探正在运行的会话，猜测正确的序列号发送包给目标主机，目标主机与攻击者建立会话，断开现有的会话。
* **基于云的物联网体系结构：**云+设备+用户管理平台
* **MQTT:** 一个应用层协议(基于OSI模型[19])，运行在TCP/IP或其他有序的、无损的、双向的连接(如WebSocket[20])上。MQTT以其轻量级设计而闻名，它适用于低带宽或不可靠网络中的资源紧张设备，因此非常适合支持物联网生态系统。
* **MQTT通信过程：**发送三个基本类型消息CONNECT、PUBLISH、SUBSCRIBE MQTT利用订阅发布模式[17]实现通信：MQTT客户端(例如物联网设备或管理应用程序)将消息发布到承载特定主题的代理，然后代理将消息路由到订阅了该主题的其他客户端
* 首先，MQTT客户机(例如智能空调或应用程序)向代理发送连接消息，用于建立MQTT会话(如果代理接受连接)。会话和客户机由一个ClientId字段(嵌入在CONNECT消息中)惟一标识，该字段类似于web会话cookie。在已建立的会话中，物联网设备订阅其相关主题(例如/DeviceId/cmd)通过向代理发送的SUBSCRIB消息(包括其主题)。代理为每个会话维护订阅状态，并将发布到主题的MQTT消息传递给它的订阅者。通过这个渠道，应用程序可以代表其用户对设备进行操作，发布对设备订阅的主题的评论(例如，开始或停止)。



* **Will 消息：**客户机可以向代理注册一个特殊的Will消息（在CONNECT类型的MQTT消息，见第二部分）为一个topic. 一旦客户端突然断开连接(即，而不是向代理发送断开连接的消息)，代理将向topic的所有订阅客户端发布will消息，允许它们采取相应的操作。
* **Retain 消息：**当MQTT客户端将消息发布到某个主题，并且没有客户端订阅该主题时，代理将简单地丢弃该消息。然而，这种简单的处理还会在客户端暂时离线时干扰发布者与其订阅客户端的通信。为了解决这个问题，MQTT客户机可以注册带有主题的保留消息(通过在常规MQTT消息中设置保留标志)，这允许代理将最后保留的消息保留在主题上，并立即将其发布给主题的任何未来订阅者。
* **PoC(全称: Proof of Concept)：**中文可译作“观点验证程序”，它主要用于证明提出者的观点是否正确，在信息安全领域这种观点一般是我们对于漏洞的判断，即目标系统是否存在漏洞。

## 应用场景

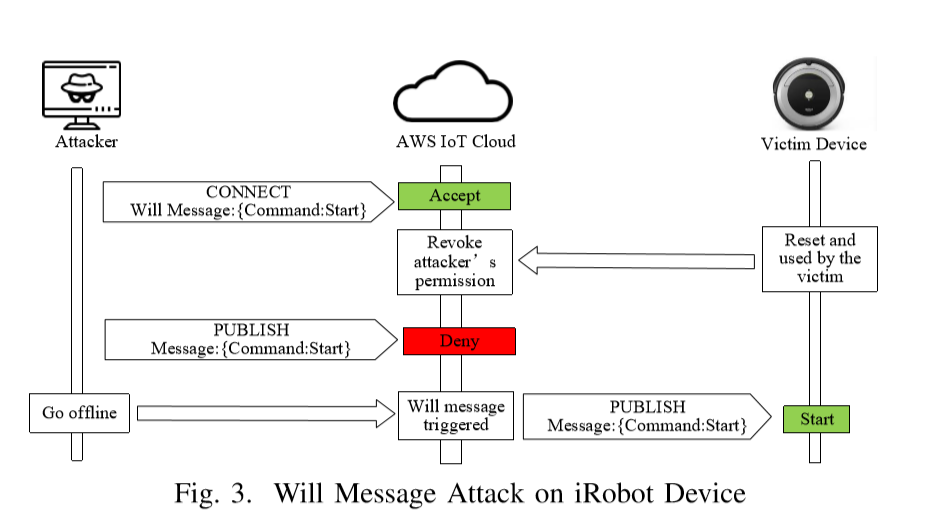
设备共享和撤销，其中一方(酒店住客、Airbnb租户、保姆等)只被信任可以临时访问物联网设备，不允许从之前和未来使用同一设备的用户那里获取信息或干扰他们的活动.

## 主要工作

### 1.未经授权的will消息

MQTT规范允许客户机向代理注册一个特殊的Will消息为一个topic, 一旦客户端突然断开连接代理将向topic的所有订阅客户端发布will消息，允许它们采取相应的操作。与其他MQTT消息一样，Will消息可以包含控制命令，也可以只包含文本。前用户可以战略性的注册一个Will消息来在之后触发当他不再有访问权限时。

### 2.通过will消息发动攻击



编写了一个脚本来注册一个Will消息，包括一个command (command:start)来启动我们的设备和代理。然后,当受害者用户(像后续客人的在同一酒店房间)重置设备和使用机器人并且我们的客户特权完全撤销(例如,拒绝发布消息到设备), 我们表明，当脚本离线时，Will消息中的命令仍然被传递并叫醒机器人。

### 3.will消息造成的原因

Will消息的契约类属性:一个主题上的已注册的Will消息有权交付给订阅该主题的所有客户端;消息是否违反订阅客户机的安全要求不是MQTT关心的问题。

### 4.未经授权的retain 消息的攻击

一个恶意的前用户利用来暗中命令一个他不再能够访问的设备，在Airbnb房间里使用过某台设备的恶意前用户可以发布一条保留的消息，其中包括任意的控制命令(例如，在凌晨3点开门)，在他仍然拥有访问权限的情况下，发布到智能锁的相关主题。稍后，当他退房并因此失去特权时，他可以等待设备重新联机。当这种情况发生时，锁将再次订阅旧主题并接收unlock命令。一旦执行，门将在凌晨3点打开，窃贼可以进入。

### 5.MQTT的会话错误

当一个客户端有一个状态改变(例如,他/她访问设备被吊销),建立了会话的状态应该更新,这是特别重要的安全等敏感的订阅状态(主题订阅)和生命周期状态(会话是否应该延迟或终止)  
 MQTT规范在更新会话状态以响应客户机的特权更改方面没有提供任何指导由此造成缺陷：只要客户端建立一个会话,订阅一个主题(例如, 酒店房间里智能扬声器的topic),甚至当用户不再允许订阅主题(例如,检查后),我们发现通过建立代理不断传递消息到客户端会话。即订阅状态的会话持续即使在用户失去了特权,这有效地使设备的恶意ex-user不断收到所有5消息生成的设备为当前用户(受害者),如个人购买历史和习惯,健康状况和数据(如心跳),等等

### 6.没有更新的会话生命周期状态

物联网云对这两个角色进行了不同的管理:将设备视为要访问的资源，将用户视为进行身份验证和授权的源泉

* 有撤销前用户访问设备的概念
* 没有撤销设备允许访问其主题的权限的概念
* 可能的攻击是：前用户获得设备证书，即使在新用户删除前用户时，攻击者仍然可以利用设备证书模拟设备并发布虚假消息给云上的主题
* Tuya物联网云如果所有者发生变化，其云下的设备凭据将被迫过期
* 针对上诉情况：只要前用户(攻击者)在凭据过期之前建立了一个会话并保持会话在线，他/她就可以代表设备通过会话向设备的主题发布虚假消息
* 当设备被重置以清除其所有现有用户以及更基本的现有状态时，所建立的MQTT会话的生命周期状态没有更新。为了安全，会话的状态应该被清理
* 发动POC攻击：step1.建立会话 step2.获取设备证书 step3.恶意会话持续并向云发布虚假消息，虚假消息将进一步发送给手机

### 7.未经身份验证的MQTT身份—没有身份验证

* ClientID劫持：针对于检测到冲突的ClientId下将客户端移动到离线状态的操作是不存在的，进行成功的攻击验证
* ClientID劫持：*MQTT*协议允许代理和客户端恢复前一会话如果客户机与先前*ClientId，*允许客户端迅速恢复之前的状态，以避免配置复杂。然而一旦将其*ClientId*公开给攻击者，这种恢复机制就可能被滥用来针对目标客户端工作
* ClientID 认证：*ClientId*是一个序列，代理要么根本不检查它的合法性，要么只检查它的基本格式
* ClientID 只确保了其唯一性没有确保其安全性，因此攻击者可以通过猜测去获取clientID
* 威胁来源：源于MQTT没有将clientID从一开始视为一个秘密
* 这种保护只能确保只有经过身份验证的平台用户才能建立MQTT连接，而不是只有授权用户才能声明连接的*ClientIds*。因此，平台上任何经过身份验证的用户都可以使用任何*ClientId*。

### 8.MQTT topic

* 由于云要同时服务大量用用户，云允许用户订阅知道的所有的MQTT主题
* 云未能正确授权使用通配符订阅MQTT主题
* 平台的任何用户都可以订阅物联网云的通用#主题，根据其在MQTT中的定义，这意味着代理上的所有MQTT主题(实际上是物联网云下的所有设备)。该漏洞会通过物联网设备在云平台上泄露所有物联网用户的各种高度关键的隐私信息
* 用户使用通配符可以绕过安全策略从而获得主题
* 恶意用户可以订阅deviceId/#并从受保护的主题接收消息。显然，AWS未能很好地解释包含通配符的资源描述，并将其与应受保护的主题关联起来。
* 设备制造商同样有问题 根据通配符描述 恶意用户可以向这个隐藏的主题发布消息，所有机器人都可以订阅该主题并接收控制命令
* POC 实验 通过简单的脚本 我们订阅平台的通用主题 获得大量的设备隐私信息 从而推断出个人信息

### 9.实验

* 8个领先的物联网云
* MQTT(3.1.1版本,当时我们的研究)的物联网通信
* 我们发现身份管理(ClientId劫持,参见III-C),消息授权(将消息和保留信息,参见III-A),会话管理(III-B节)和主题授权(III-D节)
* 身份管理：ClientId劫持攻 -- 管理MQTT ClientId方面缺乏安全和标准化的实践。 不同的格式导致clienid 劫持
* 消息授权：所有的物联网云平台，如AWS、微软、Tuya等，除了少数例外，在我们研究时，其实现并不支持两种类型的面向效用的消息(Will Message和retain Message)
* 会话管理：大多数物联网云受到不安全的会话管理(订阅和生命周期状态，参见III-B部分)的影响
* Topic 授权：主要的平台AWS和苏宁公司都受到了这类攻击。我们所研究的其他特性则不是，因为它们的当前实现没有积极利用MQTT主题中的通配符，这是一个方便的特性，考虑到当前系统设计[48]中面向可用性的趋势，安全隐患在实践中可能是不可避免的
* 结论：各云厂商在 MQTT 通信构建的不规范化

### 10.泄露的MQTT 所引发的问题

* 信息泄露引发的纵向分析

### 11.缓解措施

* 当今的物联网云无法弥补MQTT的面向可用性设计与复杂的用户设备交互的安全性需求之间的差距
* 保护的关键是添加缺少的安全模型和控制关键协议实体的设计原则
* A.管理协议标识和会话
  + - 协议层身份(例如，ClientId)(如果有的话)应该经过验证
    - 平台层身份(如Amazon帐户，参见III-C部分)对MQTT连接进行良好的身份验证
    - 建议将MQTT客户机的ClientId严格限制为逐字匹配客户机平台上的身份p\_user\_id，一种更通用的方法是维护平台标识与其允许的标识之间的映射
* B.建立面向消息的访问控制模型

通信中的客户端集合，如设备和用户。主体由其属性定义和表示。

主题的属性被指定为并且包含识别信息(id),的集合[51](例如topic), 允许发送消息到(),并且URI允许接受消息从

主体拥有权限的一组消息。

对象的属性被指定为 它包括content应用程序层信息(例如，消息内容)、URI(表示消息通道例如，消息发布到哪个MQTT主题或来自哪个MQTT主题)、source表示对象源的URI(例如，消息内容)即，创建消息的主题。

权限是主体可以持有并在对象上行使的特权。一般有两类权限，读(R)(例如，接收消息)和写(W)(例如，发布消息)。

Authorizations.授权功能评估和请求的权限，以及一组用于访问决策的授权规则。

例如：

*(1)*

* + - 在式(1)中，表示允许*Subject* *s*的权利*Right* *R* 接入到*Object* *o*(即，接收人s被允许接收一条消息o).

例如通过此模型可以有效的检查如果消息的source(o.source，例如，发送消息的用户客户端)已经失去了访问消息来源的主题的权限(通过检查不在中)，此消息(例如，*Will*消息)应该被拒绝。此外，此规则还通过检查消息的主题()是否在允许的主题集中来检查是否允许接收消息。

### 12.验证性实验

实验成功抵御攻击

## 可拓展点

* 在将一个实用的、通用的协议应用到可能涉及恶意方的领域时应该谨慎。在这种情况下，既需要评估协议不涉及的场景，也需要评估协议的各个状态，以确定协议可以保护的内容与需要保护的内容之间的差距
* 不仅应该通过适当的身份验证和授权来保护*ClientId*及其相关状态，而且还应该在适当的保护下将整个撤销过程(其安全性至关重要)添加到协议中
* 模糊的、不完整的描述或复杂的协议逻辑都可以指导我们的分析将重点放在寻找安全弱点的相关实现上 实现其自动化
* MQTT新协议仍没有在协议层方面解决我们的问题。即使它的非规范化部分提到客户端需要授权使用*ClientId*和应该提供一个实现访问控制限制客户端发布或订阅特定主题的能力,它没有提供任何建议应该如何在这种保护迫使我们研究物联网的场景
* 利用其他模型确保其他的一般消息传输协议得到有效的保护

## 总结

首次对使用物联网设备用户通信通用消息传递协议的安全风险进行了系统研究，揭示了为在简单和良好的环境下运行而设计的协议与复杂的、敌对的物联网场景之间的差距，以及用适当的安全手段来弥补这一差距所面临的挑战

## 心得

* 针对协议的分析不仅应当考虑其在当前环境下的安全性问题同时也需要评估协议的各个状态是否安全，以确定协议可以保护的内容与需要保护的内容之间的差距
* 有误导性的开发指南，促使平台的安全性脆弱
* 针对于底层通信协议的分析往往也需要检查协议各个状态的安全性问题
* 某些默认安全环境下运行的协议 在特殊的不安全环境下会存在巨大隐患
* 本篇文章体现出工作量的巨大，有相当的借鉴意义。