[rmqtt](https://github.com/rmqtt/rmqtt)

### 攻击场景与漏洞危害

考虑共享场景下的物联网应用，即智能家居系统使用MQTT协议进行物联网设备和用户管理，其中有两个用户角色。管理员（即房主）可以授权其他普通用户（例如，客人、租户）访问他的智能家居设备的权利。普通用户的访问权限可能会被撤销（租约到期）。我们认为管理员和设备是良性的，而客人可能是恶意的，会尽可能地去试图越权访问设备（越权或是维持被撤销的权限）。

##### 0x01攻击场景

首先，攻击者暂时（作为租客）拥有主题“testtopic”的发布权限。

1. 攻击者连接Broker
2. 攻击者向话题“testtopic”发布retained message，该消息被broker存储
3. 攻击者的发布权限被管理员或设备所有者撤销。
4. 智能设备上线并订阅了话题“testtopic”，此时智能设备收到攻击者发布的retained message（即使此时攻击者已经失去了发布权限）

##### 0x02漏洞危害

攻击者能够在失去发布权限之后仍然向订阅者发布消息（时机取决于新的订阅者什么时候订阅话题）。未授权访问可能导致攻击者在租约到期退房之后，仍然能向智能门锁的控制话题发布解锁命令，打开智能门锁。

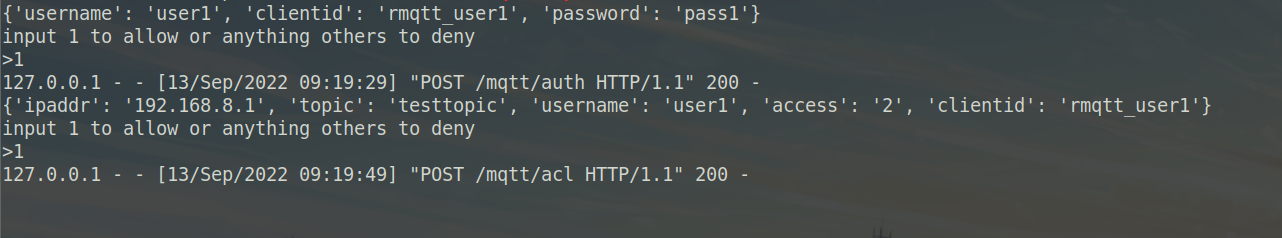
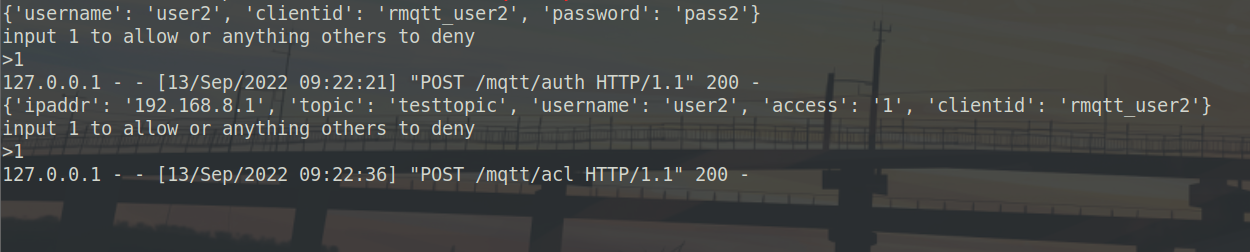
### 测试

##### 0x01测试环境

[rmqtt v0.2.3](https://github.com/rmqtt/rmqtt/releases/download/v0.2.3/rmqtt-0.2.3-x86_64-unknown-linux-musl.zip)

测试时使用rmqtt自带的rmqtt-auth-http插件<https://mqttx.app/zh>)

##### 0x02测试步骤

* user1作为攻击者连接到broker并发布一条retained message到“testtopic”，我们认为这时候攻击者具有连接和发布的权限，因此权限检查通过
* 
* 之后撤销user1的发布权限，具体操作为：后续收到任何user1的发布请求，全部拒绝操作
* user2作为智能设备，连接到broker并订阅“testtopic”，我们认为user2具有连接和订阅的权限，因此权限检查通过（注意此时只检查了user2的相关权限）
* 
* user2的客户端收到user1发布的retained message，但是此时user1已经不具备发布权限
* 