物联网安全课程实验报告

**实验三**



|  |  |  |
| --- | --- | --- |
| **实验名称** | ： | 物联网设备加密通信设计与实现 |
| **姓名** | ： | 辛杰 |
| **小组** | ： | 田博仁-王梓骁-辛杰 |
| **学号** | ： | 2213034 |
| **专业** | ： | 物联网工程 |
| **提交日期** | ： | 2024年11月1日 |

## 一、实验目的

了解目前主流的基于云的物联网通信原理，学会使用基本的密码学工具，并在消费物联网应用场景下构建安全加密通信方案。

## 二、实验要求及要点

•实验目标

•组员分工情况（须有一位组员专门从攻击者视角做安全评估）

•方案设计

•方案实现（包括网络拓扑，实现技术细节，功能效果演示，等等）

•系统方案安全性评估（包括系统设计的不足和未来改进思路等）

•每位成员的收获与感悟，体会“红蓝”对抗对构建安全系统的作用

•提交源代码

## 三、组员分工

辛杰：消息设计与文本处理、加解密测试、整合测试

田博仁：MQTT服务器本地测试、MQTT服务器网络测试

王梓骁：安全评估

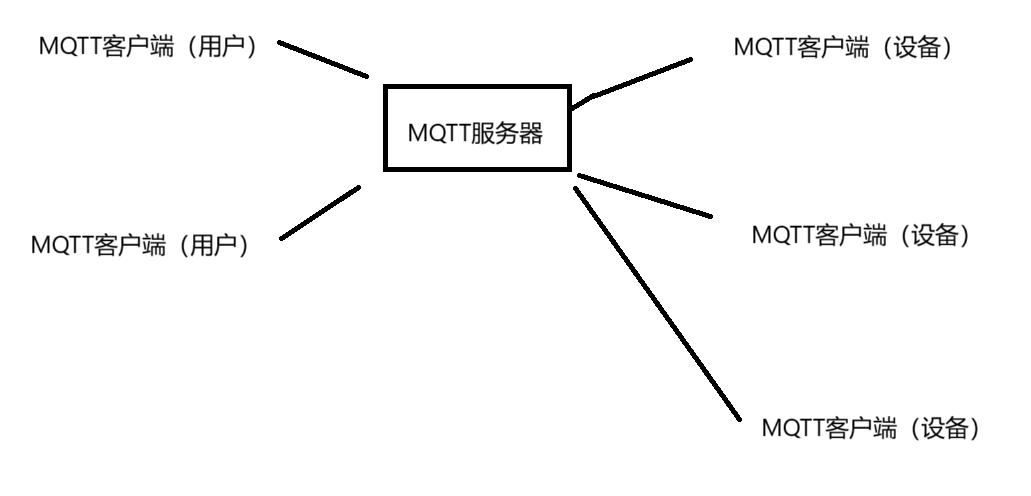
## 四、实验内容

### 1、方案设计

本方案设计了一个基于MQTT协议的智能家居原型系统，该系统通过模拟多个用户和物联网设备（包括空调、灯泡和插座）的交互，实现了设备控制、状态查看的核心功能。系统采用时间戳和nonce机制和UID等来确保通信安全，抵御指令重放攻击，并从不同攻击者的视角对系统进行了全面的安全评估，以确保系统的安全性和可靠性。

### 2、方案实现

#### 1）网络拓扑



说明：

* MQTT服务器地址：192.168.188.130
* 直线代表可以subscribe和publish
* 设备订阅的主题为home/相应的device\_id
* 用户订阅的主题为home/#（表示订阅home下面的全部topic）

#### 2）实现技术细节

##### 用户端

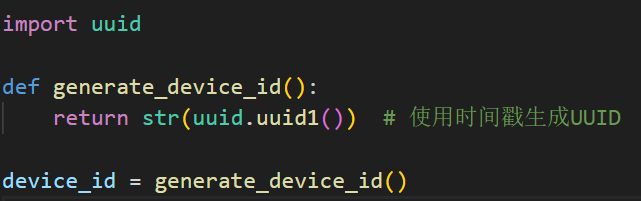
* 时间戳生成： 使用datetime.now().strftime("%Y-%m-%d %H:%M:%S")生成当前时间的时间戳，格式为年-月-日 时：分：秒。这个时间戳用于消息的第一部分，确保消息的时效性。
* 随机数（nonce）生成： 使用random.randint(100000000, 999999999)生成一个介于1亿到9亿之间的随机整数，作为消息的第二部分。这个随机数用于增加消息的唯一性，防止重放攻击。
* 消息构造： 将动作（action）、时间戳（timestamp）和随机数（nonce）通过管道符|连接起来，形成一条完整的消息。这条消息将被发送到MQTT服务器。
* SHA-256哈希计算： 使用hashlib.sha256(message.encode())对构造的消息进行SHA-256哈希计算。SHA-256是一种安全的哈希算法，可以确保消息的完整性和不可篡改性。
* 哈希值转换： 将计算得到的哈希值通过hexdigest()方法转换为十六进制字符串，方便后续的传输和比较。



通过时间戳、随机数和哈希值的组合，使得每条消息都是唯一的，并且可以被接收方验证。

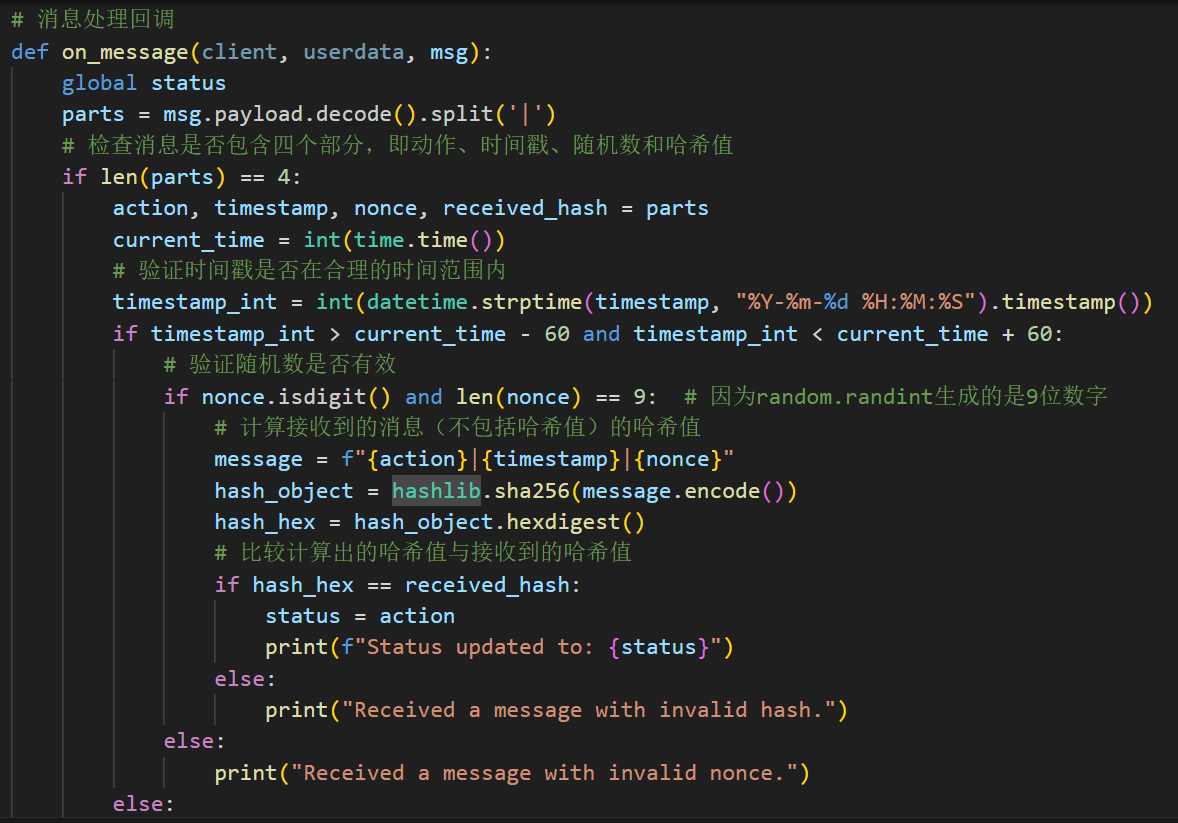
##### 设备端

###### 1.使用当前时间戳和随机数生成唯一的UUID。



###### 2.消息验证

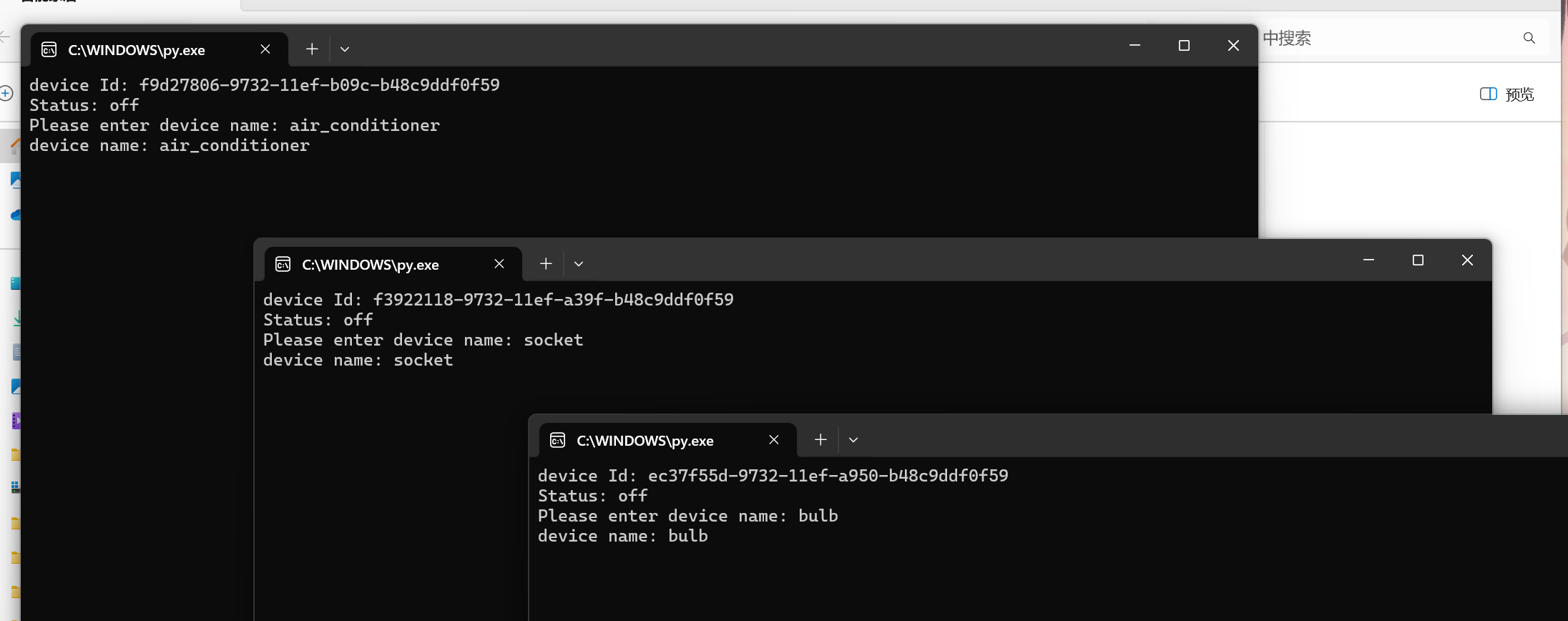
* 消息格式验证： 通过`if len(parts) == 4:`检查消息是否包含四个部分，这是为了确保消息格式正确。
* 时间戳验证：通过比较当前时间（`current\_time`）与消息中的时间戳（`timestamp\_int`），验证时间戳是否在合理的时间范围内（前后60秒内）。
* 随机数（nonce）验证：使用`nonce.isdigit()`检查随机数是否为数字。使用`len(nonce) == 9`检查随机数是否为9位数字，这是因为`random.randint(100000000, 999999999)`生成的随机数是9位的。
* 哈希值验证：重新计算接收到的消息（不包括哈希值）的SHA-256哈希值。将计算出的哈希值（`hash\_hex`）与接收到的哈希值（`received\_hash`）进行比较，以验证消息的完整性和真实性。



#### 3）功能效果演示

##### 支持至少2个用户，至少3个物联网设备（空调、灯泡 、插座不同类型）同时在线

3个设备



2个用户（一个本机用户，一个虚拟机登录用户）

本机用户

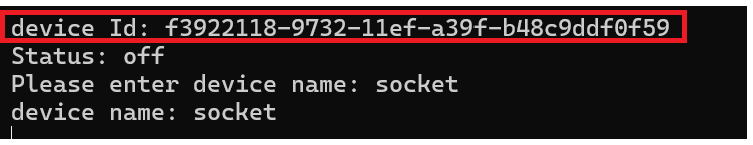


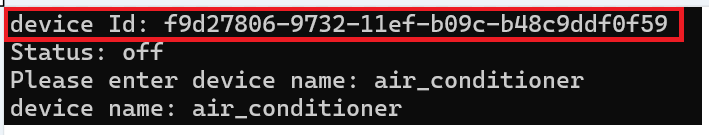
虚拟机Ubuntu登录用户

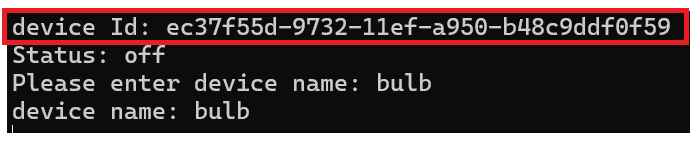


##### 设备绑定

每次运行device.py即可绑定唯一device id

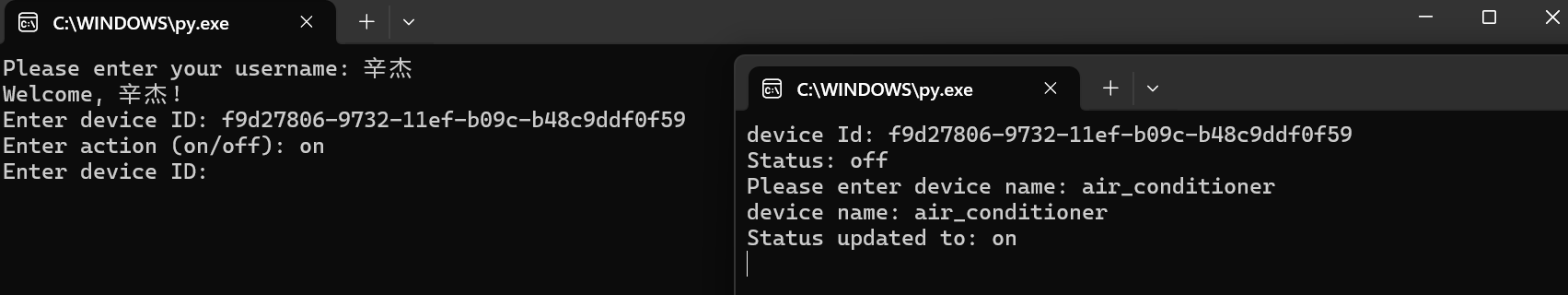




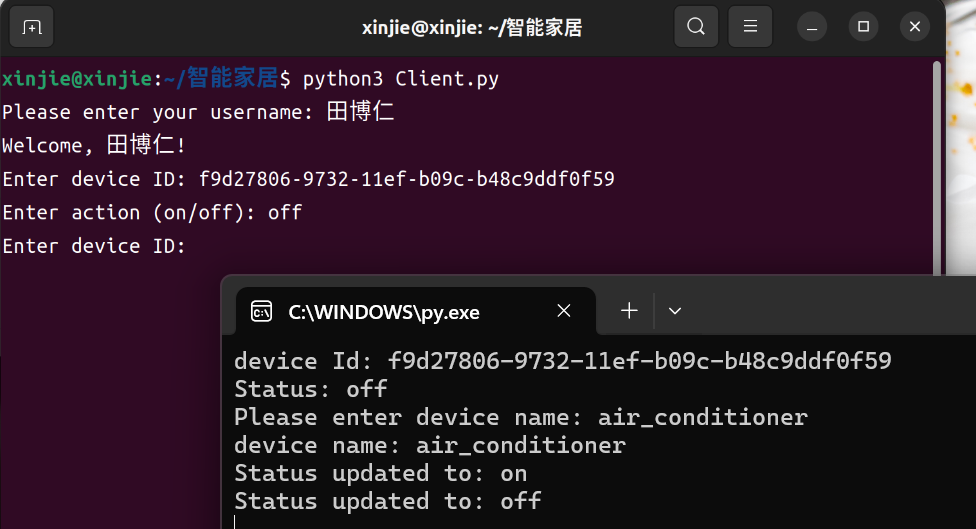


##### 设备控制

每个用户只要知道device id，只需输入id，然后输入状态，相应的设备就会就会改变状态，允许多个用户同时操作任意个设备。



虚拟机上用户也能控制这个冰箱



#### 3、系统方案安全性评估

## 五、回答问题

### 1）智能家居设备的使用与工业控制系统面临风险有何差异？

1. 网络攻击的目标和影响：

智能家居设备的风险更多关注于个人隐私泄露和家庭安全。例如，数据传输未加密可能导致用户信息泄露，以及远程控制命令缺乏加固授权，存在非法入侵、劫持应用的风险。

工业控制系统面临的风险则更侧重于生产安全和运营中断。例如，缺少必要的监测预警手段，工业数据面临被窃取、篡改、丢失等安全风险，以及直接对PLC、DCS等控制器下发错误指令，可能导致生产事故。

1. 技术复杂性和专业性：

智能家居设备通常面向普通消费者，技术门槛相对较低，用户可能缺乏必要的安全知识和意识。

工业控制系统则更为复杂和专业，需要专业知识和技能来维护和操作，安全风险往往与系统的专业操作和维护不当有关。

1. 安全防护能力：

智能家居设备的安全防护能力通常较弱。例如，客户端APP未安全检测，存在代码漏洞，硬件设备存在调试接口，使用有安全漏洞的操作系统或第三方库。

工业控制系统虽然在原理上基本符合一般计算机或通信系统的通信原理，但也存在一些特殊性，如对数据传输的实时性极其敏感，通信网络上的任何干扰都可能产生影响。

1. 网络通信漏洞风险：

智能家居设备可能因为网络通信漏洞而面临风险，例如缺少防护的网络边界风险。

工业控制系统可能因为现场控制设备性能的限制和实时性的要求，ICS通信网络上数据传输通常不加密或采用简单的加密算法，很容易破解，存在信息泄露或数据被伪造的风险。

1. 管理制度和责任：

智能家居设备的安全管理制度可能不够完善，管理不到位、责任不清晰。

工业控制系统则需要更为严格的管理制度和责任划分，以确保生产安全和连续性。

### 2）列举几种可用来实现加密通信的常见密码算法，并进行对比分析

1. AES（高级加密标准）

类型：对称加密算法。

特点：AES是目前安全强度较高、应用范围较广的对称加密算法，支持128、192、256位的密钥长度。它以分组整体替换的方式工作，每次循环处理更多的原文数据，即使需要密钥参与运算，也能保持较高的速度。

安全性：在只有密文的情况下，需要穷举密钥进行解密，最长得解密2^256次才破解。

2. RSA

类型：非对称加密算法。

特点：RSA算法使用一对密钥，即公钥和私钥。公钥可以公开，而私钥必须保密。它适用于加密少量数据或用于数字签名。RSA在数据量少时计算量小，但数据量大时，运算量大，速度较慢。

安全性：密钥长度有512、1024、2048位等，破解需要穷举密钥，运行2^512次到2^2048次，破解难度极高。

3. DES（数据加密标准）

类型：对称加密算法。

特点：DES算法的密钥长度为56位，由于密钥长度较短，安全性较低，目前已逐渐被淘汰。

安全性：需要穷举密钥解密，需要2^56次尝试才能破解。

4. MD5（消息摘要算法第五版）

类型：摘要算法。

特点：MD5是一种广泛使用的哈希函数，产生128位的哈希值。它速度快，但由于雪崩效应和已知的碰撞弱点，不再推荐用于安全敏感的应用。

安全性：不可逆，破解只能靠穷举法，需要2^128次尝试。

5. SHA-256（安全哈希算法256位）

类型：摘要算法。

特点：SHA-256是SHA-2算法家族中的一员，产生256位的哈希值。它比MD5更安全，常用于文件或数据校验，保存密码。

安全性：不可逆，破解只能靠穷举法，需要2^256次尝试。

对比分析：

性能：AES和MD5在速度上表现较好，而RSA由于其算法复杂性，在处理大量数据时速度较慢。

安全性：RSA和SHA-256因其较长的密钥长度和哈希值，提供了更高的安全性。DES由于密钥长度较短，安全性较低，已逐渐被淘汰。AES提供了较好的安全性和速度的平衡。

应用场景：AES适合加密大量数据，RSA适合加密少量数据或数字签名，MD5和SHA-256常用于数据完整性校验。

### 3）什么是虚拟机的NAT模式、桥接模式、Host-only模式？

NAT模式（网络地址转换模式）：

在NAT模式下，虚拟机通过宿主机进行网络地址转换（NAT）来访问外部网络。宿主机相当于一个路由器，虚拟机的网络流量会通过宿主机的网络接口，并且虚拟机的IP地址会被宿主机的IP地址所替代。

虚拟机可以访问外部网络，但是外部网络无法直接访问虚拟机。

这种模式适合开发和测试环境，因为它提供了一定程度的隔离和安全性。

桥接模式（Bridged模式）：

桥接模式下，虚拟机相当于直接连接到物理网络中的一个独立的设备。虚拟机将在网络上获得一个独立的IP地址，就像宿主机上的一个物理设备一样。

虚拟机可以与同一网络中的其他设备进行通信，包括宿主机和其他虚拟机，就像它们都是连接到同一个交换机的不同设备一样。

这种模式适合需要虚拟机完全融入物理网络环境的场景，例如，当虚拟机需要直接访问局域网中的其他设备或服务时。

Host-only模式（仅宿主机模式）：

在Host-only模式下，虚拟机只能与宿主机进行网络通信，而不能直接访问外部网络。虚拟机和宿主机之间的通信是通过一个私有的虚拟网络进行的。

这种模式通常用于那些不需要访问外部网络，但需要与宿主机进行文件共享或其他网络服务的场景。

虚拟机之间也可以通过宿主机进行通信，但它们不会暴露给外部网络。

## 六、收获感悟

辛杰：

田博仁：

王梓骁：