

# Modbus流量捕获（环境搭建 + 流量监听）

## 0x00 前言

Modbus是较为通用的工控系统协议。本文所捕获的流量来自于“主站设备仿真软件” ModbusPoll 以及“从站设备仿真软件” ModbusSlave 的通信过程；本文通过 Ettercap 和 Wireshark 进行流量监听。

## 0x01 环境搭建

为了进行实验，进行如下环境搭建：

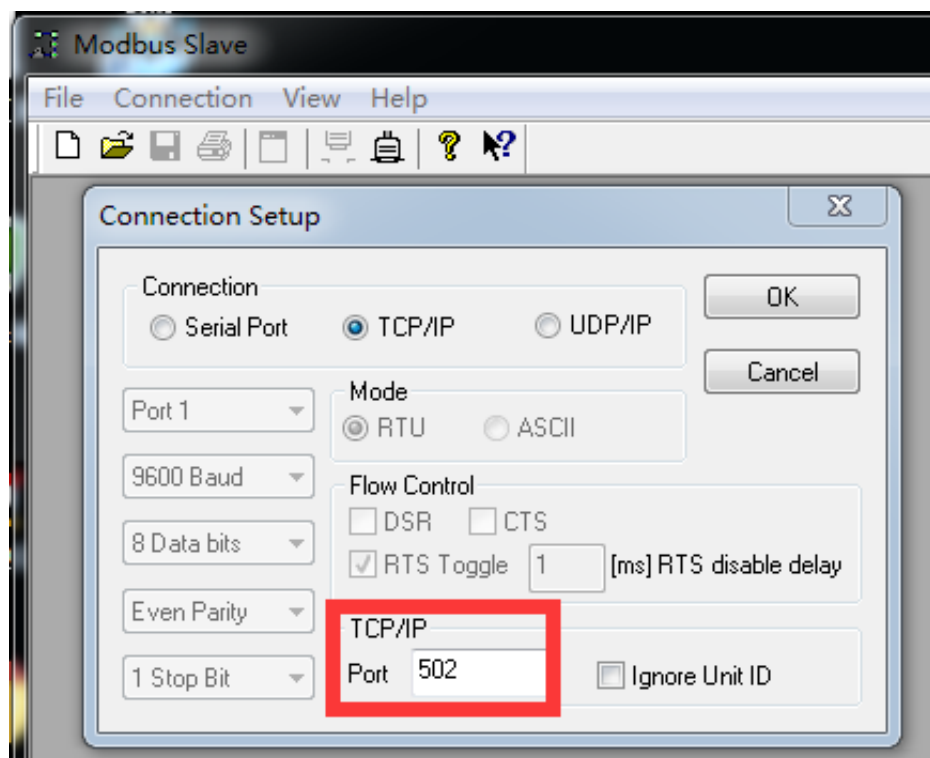
- 主站设备：Win 10 虚拟机，IP：192.168.1.147，安装有 ModbusPoll；
- 从站设备：Win 7，IP：192.168.1.147，安装有 ModbusSlave；
- 攻击机：Kali Linux 虚拟机，IP：192.168.1.148；

在安装 ModbusPoll 和 ModbusSlave 时，主要参考了文章《[ModbusPoll及ModbusSlave安装及使用指南](#)》。

在安装好 ModbusPoll 与 ModbusSlave 后，建立“主站设备”与“从站设备”的通信。

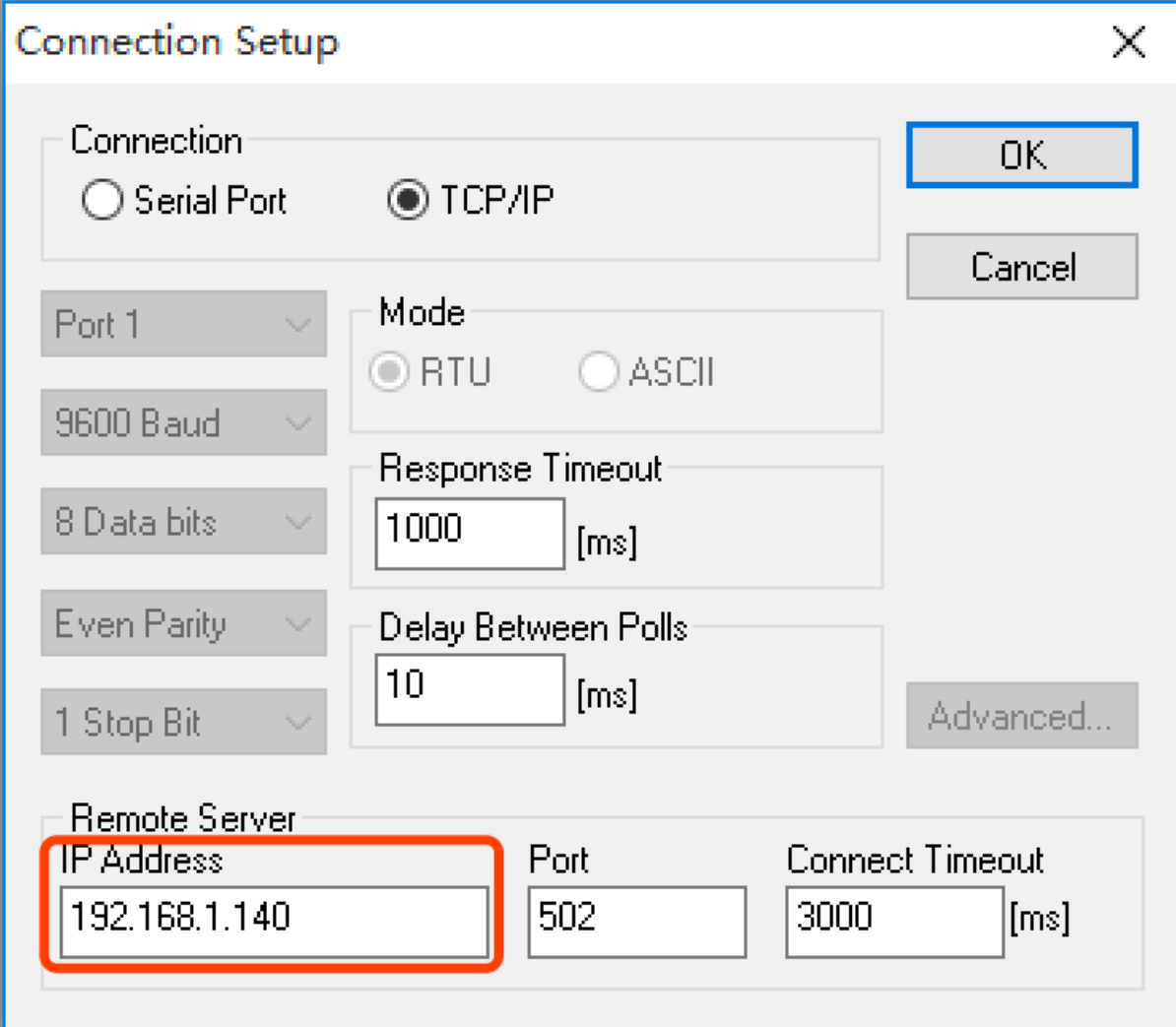
- 从站设备

进入【Connection】 / 【Connect】 进行如下配置：



- 主站设备

进入【Connection】 / 【Connect】 进行如下配置：



The image shows the 'Modbus Poll' application window with the 'Connection Setup' dialog box open. The dialog box has a title bar with a close button (X). It contains several sections for configuring the connection. The 'Connection' section has two radio buttons: 'Serial Port' and 'TCP/IP', with 'TCP/IP' selected. To the right of this section are 'OK' and 'Cancel' buttons. Below this, there are several groups of settings. The first group on the left has five dropdown menus: 'Port 1', '9600 Baud', '8 Data bits', 'Even Parity', and '1 Stop Bit'. The second group has two radio buttons: 'RTU' (selected) and 'ASCII'. The third group has a 'Response Timeout' field set to '1000' [ms]. The fourth group has a 'Delay Between Polls' field set to '10' [ms]. To the right of these groups is an 'Advanced...' button. At the bottom, there is a 'Remote Server' section with three fields: 'IP Address' (set to '192.168.1.140'), 'Port' (set to '502'), and 'Connect Timeout' (set to '3000' [ms]). The 'IP Address' field is highlighted with a red rectangle.

Modbus Poll

File Connection View Help

Connection Setup

Connection

☐ Serial Port ☒ TCP/IP

Port 1

9600 Baud

8 Data bits

Even Parity

1 Stop Bit

Mode

☒ RTU ☐ ASCII

Response Timeout

1000 [ms]

Delay Between Polls

10 [ms]

Advanced...

Remote Server

IP Address

192.168.1.140

Port

502

Connect Timeout

3000 [ms]

OK

Cancel

也可以在主站设备通过点击【Display】 / 【Communication...】来监控报文：

Modbus Poll - Mbpoll1

File Edit Connection Setup Functions **Display** View Window Help

Tx = 34: Err = 0: ID = 1: F = 03: SR = 1000ms

	Alias	00000
0		0
1		0
2		0

Communication Traffic

Exit Stop Save Copy ☐ Stop on Error

```
000030-Tx:00 14 00 00 00 06 01 03 00 00 00 0A
000031-Rx:00 14 00 00 00 17 01 03 14 00 00 00
000032-Tx:00 15 00 00 00 06 01 03 00 00 00 0A
000033-Rx:00 15 00 00 00 17 01 03 14 00 00 00
000034-Tx:00 16 00 00 00 06 01 03 00 00 00 0A
000035-Rx:00 16 00 00 00 17 01 03 14 00 00 00
000036-Tx:00 17 00 00 00 06 01 03 00 00 00 0A
000037-Rx:00 17 00 00 00 17 01 03 14 00 00 00
000038-Tx:00 18 00 00 00 06 01 03 00 00 00 0A
000039-Rx:00 18 00 00 00 17 01 03 14 00 00 00
000040-Tx:00 19 00 00 00 06 01 03 00 00 00 0A
000041-Rx:00 19 00 00 00 17 01 03 14 00 00 00
000042-Tx:00 1A 00 00 00 06 01 03 00 00 00 0A
000043-Rx:00 1A 00 00 00 17 01 03 14 00 00 00
000044-Tx:00 1B 00 00 00 06 01 03 00 00 00 0A
000045-Rx:00 1B 00 00 00 17 01 03 14 00 00 00
000046-Tx:00 1C 00 00 00 06 01 03 00 00 00 0A
000047-Rx:00 1C 00 00 00 17 01 03 14 00 00 00
000048-Tx:00 1D 00 00 00 06 01 03 00 00 00 0A
000049-Rx:00 1D 00 00 00 17 01 03 14 00 00 00
000050-Tx:00 1E 00 00 00 06 01 03 00 00 00 0A
000051-Rx:00 1E 00 00 00 17 01 03 14 00 00 00
000052-Tx:00 1F 00 00 00 06 01 03 00 00 00 0A
000053-Rx:00 1F 00 00 00 17 01 03 14 00 00 00
000054-Tx:00 20 00 00 00 06 01 03 00 00 00 0A
000055-Rx:00 20 00 00 00 17 01 03 14 00 00 00
000056-Tx:00 21 00 00 00 06 01 03 00 00 00 0A
000057-Rx:00 21 00 00 00 17 01 03 14 00 00 00
```

## 0x02 流量监听

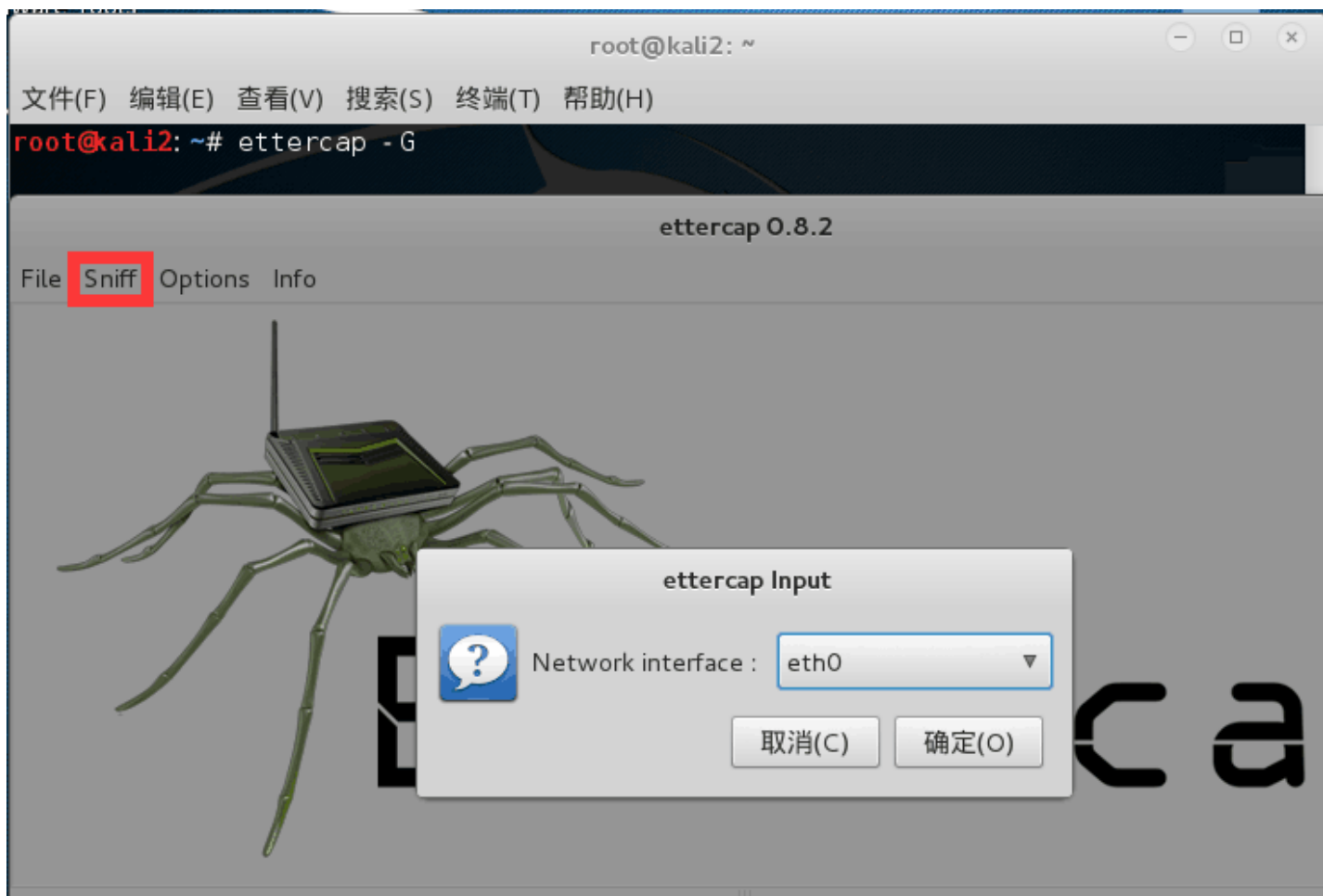
在流量监听时，主要参考了文章《<https://blog.csdn.net/u013752202/article/details/78568995>》

- 在 Kali Linux 虚拟机运用 `Ettercap` 做流量牵引：

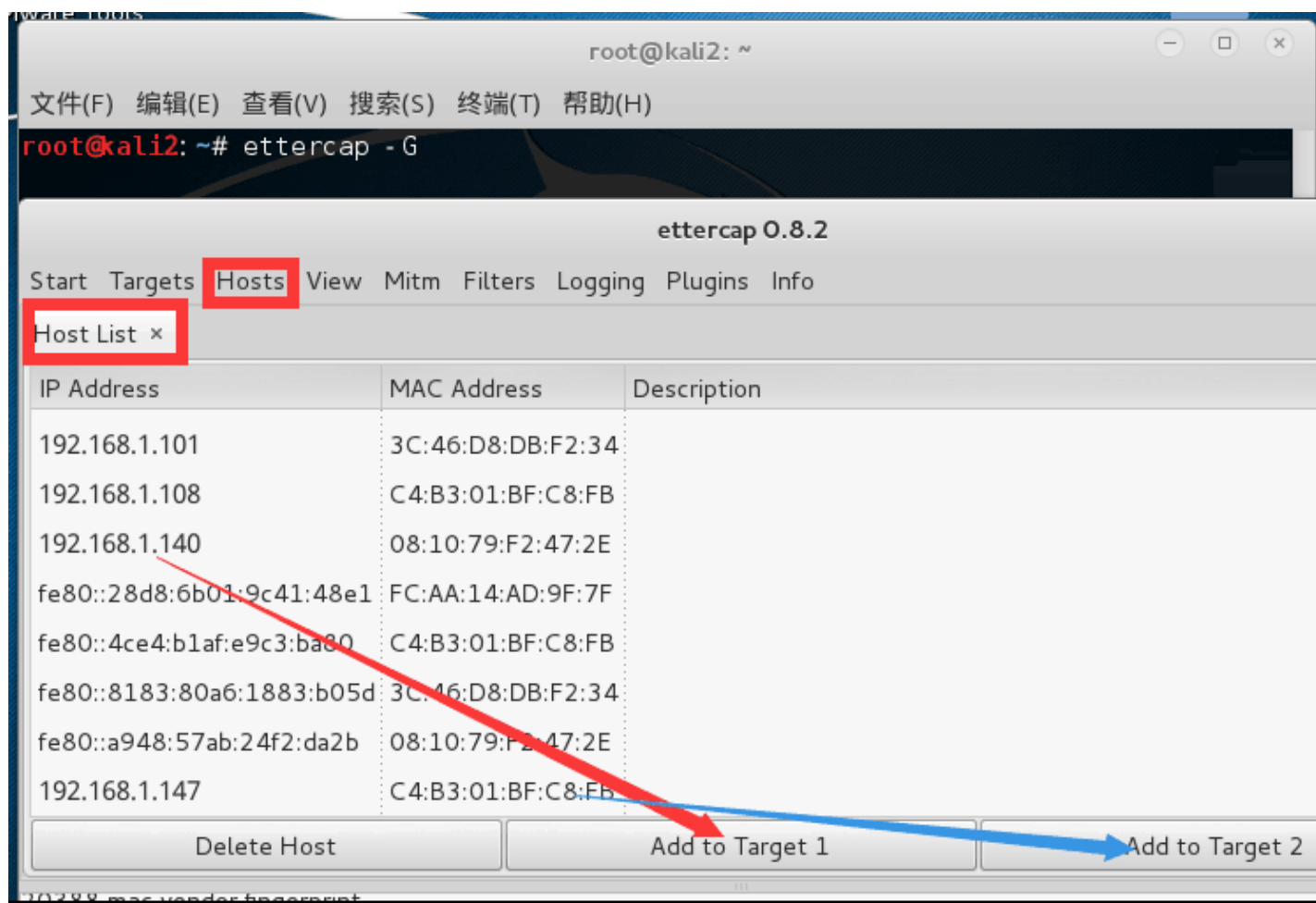
执行如下命令以打开ettercap：

```
ettercap -G
```

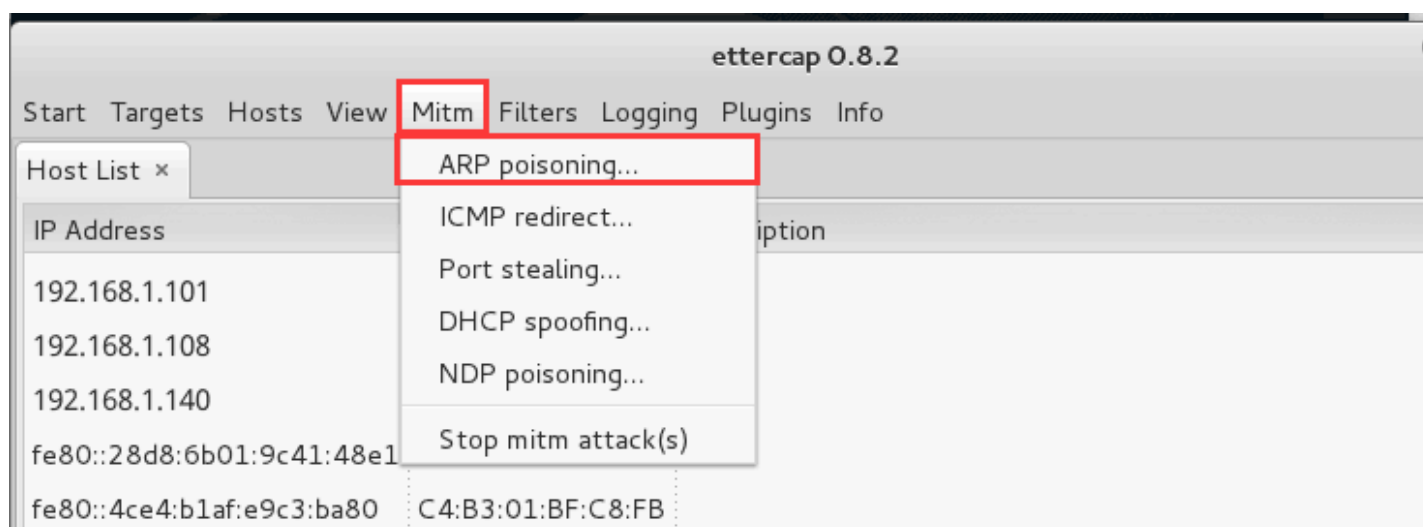
选择【Sniffer】/【Unified sniffing】以及相对应的网卡：



选择【Hosts】/【Hosts list】，查看局域网下的主机列表，并进行“将从站设备 Add to Target 1”与“将主站设备 Add to Target 2”的操作：



接着进行“ARP毒化”（ARP欺骗）：



选择“Sniffer remote connections”：



使用 Wireshark 进行辅助分析。

Capturing from eth0 [Wireshark 1.12.6 (Git Rev Unknown from unknown)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear

Time	Source	Destination	Protocol	Length	Info
13	1.154942000	192.168.1.140	192.168.1.147	Modbus/T	83 Respons
14	1.205405000	192.168.1.147	192.168.1.140	TCP	60 60456-5

13 1.154942000 192.168.1.140 192.168.1.147 Modbus/TCP 83 Response: Trans: 2

Frame 13: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface eth0

Ethernet II, Src: 08:10:79:f2:47:2e (08:10:79:f2:47:2e), Dst: c4:b3:01:bf:c4

Internet Protocol Version 4, Src: 192.168.1.140 (192.168.1.140), Dst: 192.168.1.147 (192.168.1.147)

Transmission Control Protocol, Src Port: 502 (502), Dst Port: 60456 (60456)

Modbus/TCP

Modbus

Function Code: Read Holding Registers (3)

Byte Count: 20

Register 0 (UINT16):	0
Register 1 (UINT16):	0
Register 2 (UINT16):	0
Register 3 (UINT16):	0
Register 4 (UINT16):	0
Register 5 (UINT16):	0
Register 6 (UINT16):	0
Register 7 (UINT16):	0
Register 8 (UINT16):	0

Modbus Poll - Mbpoll1

File Edit Connection Setup Functions Display View

05 06 15 16 22

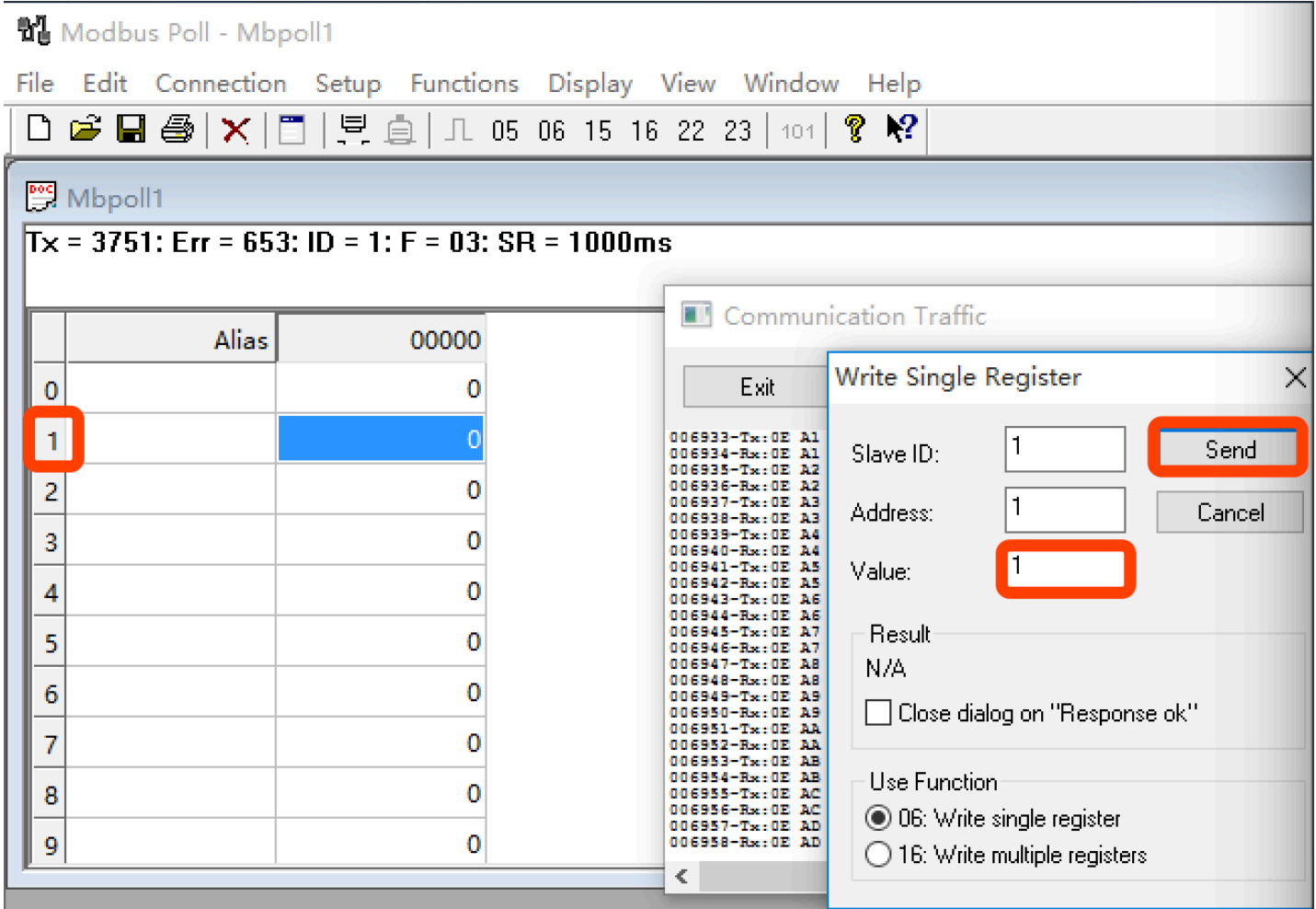
Mbpoll1

Tx = 3136: Err = 646: ID = 1: F = 03: SR = 1000ms

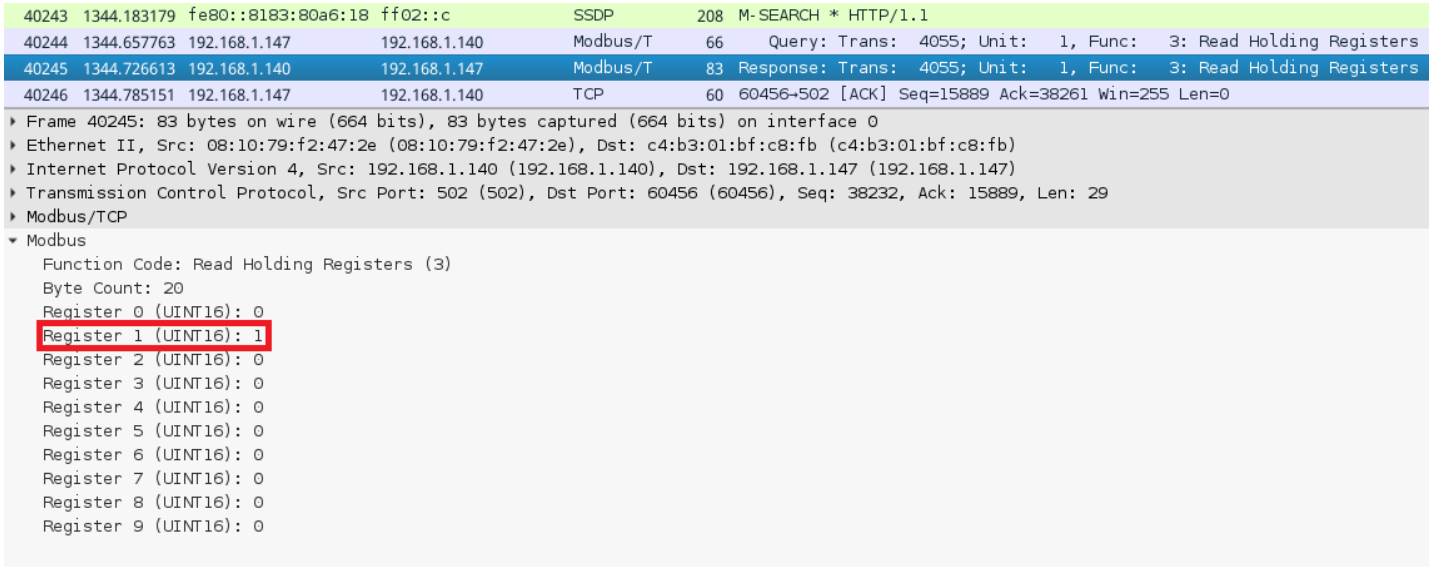
	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

对比“Wireshark的报文”与“Modbus Poll对寄存器设置的值”，可以它们均为“0，0，0，0，0，0，0，0，0”，对应起来了。

为了验证，在“Modbus Poll”将寄存器的值设置为“0，1，0，0，0，0，0，0，0”：



可以发现，“Wireshark的报文”中的寄存器的值也发生了相应的变化。



### 0x03 反思

通过本文的实践，可以对Modbus流量进行捕获。但为了对Modbus线圈和寄存器进行写操作，可能要借助如下方法（有待探究）：

- 运用Python的Scapy模块对数据包进行操作；
- 运用Modbus-cli(<https://github.com/tallakt/modbus-cli>)对线圈和寄存器进行读写；
- 运用Modbus VCR (<https://github.com/reidmefirst/modbus-vcr>) 与ettercap工具记录Modbus协议的流量并进行重放；
- 运用Metasploit的模块终止CPU运行  
([https://www.rapid7.com/db/modules/auxiliary/admin/scada/modicon\\_command](https://www.rapid7.com/db/modules/auxiliary/admin/scada/modicon_command)) 。

参考书籍：《黑客大曝光 工业控制系统安全》