(2) Analysis of S7COMM protocol

tags: Industrial control

table of Contents

Foreword

The last article describes the basic principles and structure of the Modbus protocol. This article shifts your eyes to the private agreement to see another giant Siemens S7COMM.

What is S7COMM?

Siemens is a super large company in Germany, has its figure in energy, industrial, medical, infrastructure, etc., and it is also a 66th global top 500. As a large businesses that are based on telegraphs, it is more important to communicate, S7COMM is a exclusive agreement designed by Siemens for communication between the PLCs produced, the communication between SCADA and PLC.

Unlike Modbus application layer protocol, S7COMM's protocol stack modification is higher. After the application layer organization, after further processing of the TPKT protocol, the TPKT protocol is further processed, and the following is the S7COMM protocol given by Wireshark Wiki. Stack:

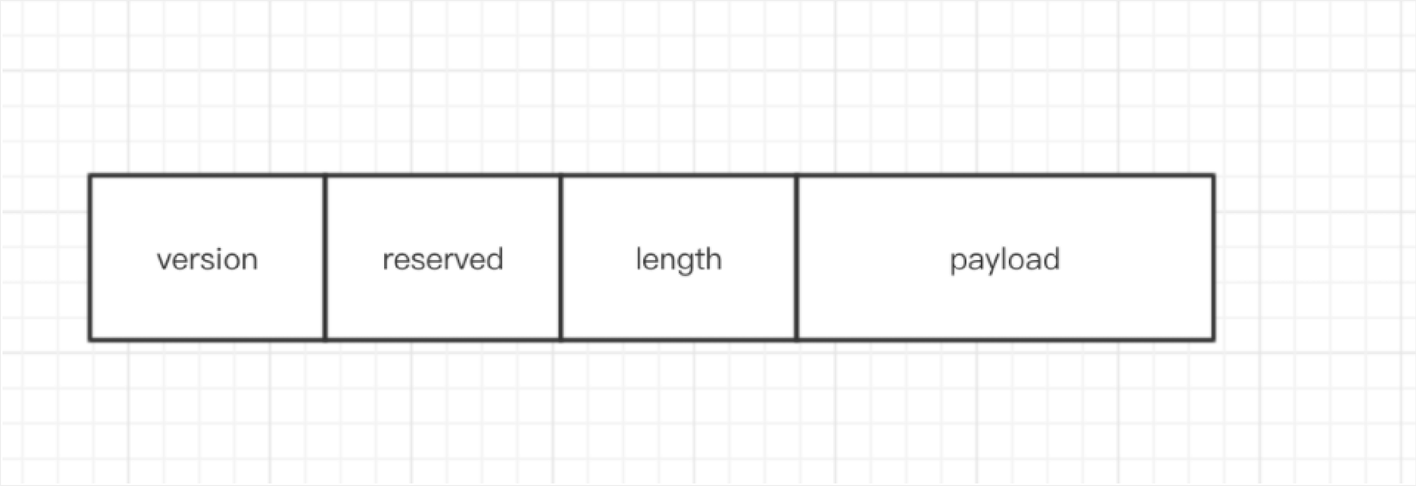
OSI Layer	Protocol
Application Layer	S7 communication
Presentation Layer	S7 communication（COTP）
Session Layer	S7 communication （TPKT）
Transport Layer	ISO-on-TCP（RFC 1006）
Network Layer	IP
Data Link Layer	Ethernet
Physical Layer	Ethernet

In view of the transfer of the packet logic, the encapsulation of the high-rise is transferred to the lower layer, but after we receive the package is the low-level layer disassembly to the upper layer, based on the reverse thinking, the analysis after we will be low-directed Highly launched.

TPKT protocol

I believe that everyone should be more familiar with the content of the transport layer, all the basic content of TCP / IP, I will not repeat it, directly from the session layer.

The TPKT protocol is a transport service protocol that transitions for the upper COPT and the lower TCP. Our common RDP protocol (Remote Desktop Protocol, Windows Remote Desktop Protocol) is also TPKT, the default TCP port of TPKT is 102 (RDP 3389), in fact, it is not much data added to PayLoad, mainly the following a:

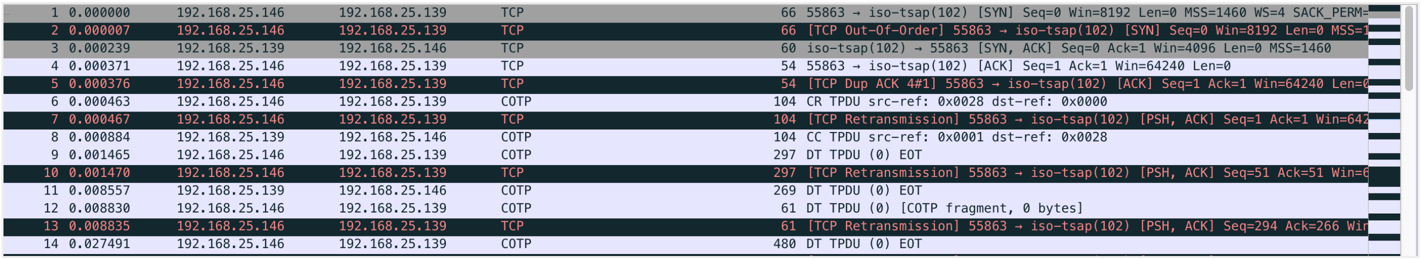


- Version, 1byte, indicating version information
- Reserved, 1byte, seeing this name, knowing it is preserved.
- Length, 2byte, the total length of PayLoad and these three parts

COTP protocol

The full name of the COTP protocol is Connection-Oriented Transport Protocol, which is a connected transport protocol. It can be seen from this name. Its transmission is inevitably dependent on the connection, so there must be a TCP handshake to establish links before transferring data. .

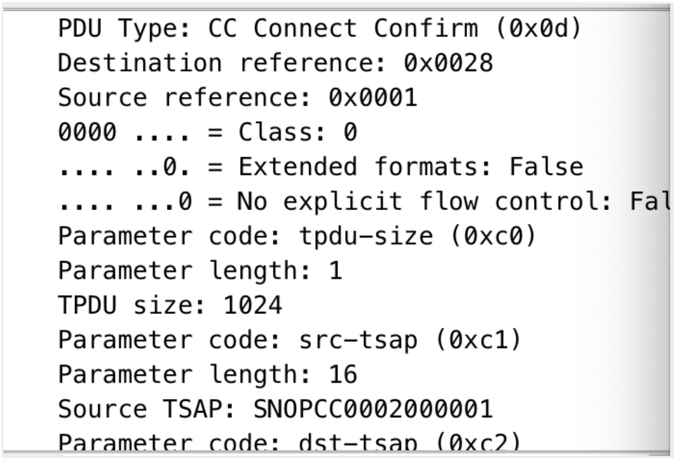
Let's take a look at the specific traffic package



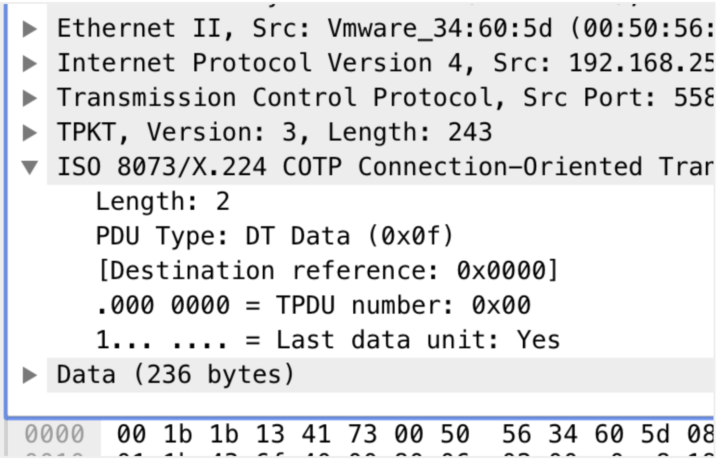
```
 1 0.000000   192.168.25.146   192.168.25.139   TCP     66  55863 → iso-tsap(102) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=
 2 0.000007   192.168.25.146   192.168.25.139   TCP     66  [TCP Out-Of-Order] 55863 → iso-tsap(102) [SYN] Seq=0 Win=8192 Len=0 MSS=1
 3 0.000239   192.168.25.139   192.168.25.146   TCP     60  iso-tsap(102) → 55863 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0 MSS=1460
 4 0.000371   192.168.25.146   192.168.25.139   TCP     54  55863 → iso-tsap(102) [ACK] Seq=1 Ack=1 Win=64240 Len=0
 5 0.000376   192.168.25.146   192.168.25.139   TCP     54  [TCP Dup ACK 4#1] 55863 → iso-tsap(102) [ACK] Seq=1 Ack=1 Win=64240 Len=0
 6 0.000463   192.168.25.146   192.168.25.139   COTP   104  CR TPDU src-ref: 0x0028 dst-ref: 0x0000
 7 0.000467   192.168.25.146   192.168.25.139   TCP    104  [TCP Retransmission] 55863 → iso-tsap(102) [PSH, ACK] Seq=1 Ack=1 Win=642
 8 0.000884   192.168.25.139   192.168.25.146   COTP   104  CC TPDU src-ref: 0x0001 dst-ref: 0x0028
 9 0.001465   192.168.25.146   192.168.25.139   COTP   297  DT TPDU (0) EOT
10 0.001470   192.168.25.146   192.168.25.139   TCP    297  [TCP Retransmission] 55863 → iso-tsap(102) [PSH, ACK] Seq=51 Ack=51 Win=6
11 0.008557   192.168.25.139   192.168.25.146   COTP   269  DT TPDU (0) EOT
12 0.008830   192.168.25.146   192.168.25.139   COTP    61  DT TPDU (0) [COTP fragment, 0 bytes]
13 0.008835   192.168.25.146   192.168.25.139   TCP     61  [TCP Retransmission] 55863 → iso-tsap(102) [PSH, ACK] Seq=294 Ack=266 Wir
14 0.027491   192.168.25.146   192.168.25.139   COTP   480  DT TPDU (0) EOT
```

The first is the three handshakes of TCP, and the TCP connection is established between 192.168.25.146 and 192.168.25.139, which is then the package of COTP. Note that here Wireshark is marked with CR and CC, and the COTP package is DT, The CR and CCs here are actually CONNECT REQUEST and CONNET CONFIRM, which is to establish a connection process. After the connection is established, the DT package is sent, which is DATA, which is transmitted.

Let's take a look at the data they carry.



```
PDU Type: CC Connect Confirm (0x0d)
Destination reference: 0x0028
Source reference: 0x0001
0000 .... = Class: 0
.... ..0. = Extended formats: False
.... ...0 = No explicit flow control: Fal
Parameter code: tpdu-size (0xc0)
Parameter length: 1
TPDU size: 1024
Parameter code: src-tsap (0xc1)
Parameter length: 16
Source TSAP: SNOPCC0002000001
Parameter code: dst-tsap (0xc2)
```



```
▶ Ethernet II, Src: Vmware_34:60:5d (00:50:56:
▶ Internet Protocol Version 4, Src: 192.168.25
▶ Transmission Control Protocol, Src Port: 558
▶ TPKT, Version: 3, Length: 243
▼ ISO 8073/X.224 COTP Connection-Oriented Trar
    Length: 2
    PDU Type: DT Data (0x0f)
    [Destination reference: 0x0000]
    .000 0000 = TPDU number: 0x00
    1... .... = Last data unit: Yes
▶ Data (236 bytes)

0000  00 1b 1b 13 41 73 00 50  56 34 60 5d 08
```

It can be seen that the DT package and the connecting package have obvious differences. The connection package has a lot of content, which is actually two forms of COPT packages, the COTP connection package (COTP FUCTION PACKET)

First, look at the COPT connection package, we can see the above Wireshark, we can see that there are several fields:

- Length, 1byte, data length, but does not contain Length this field (personal feeling is very strange ...)
- PDU Type, 1 Byte, Identification Type, 0x0d in the diagram is the type of connection confirmation, and there is often

```
0xE, connection request
 0x0d, connection confirmation
 0x08, disconnection request
 0x0c, disconnection confirmation
 0x05, refused
```

- DST Reference, 2byte, reference, can be considered to be the only identification target
- SRC Reference, 2Byte, source reference, same
- Option, 1byte, you can see that Wireshark is removed for the first four and after two digits: the first four digits Class, that is, the second bit of the logo category

corresponds to Extended Formats, whether to use the expansion of the top 4 correspondence No expected NO EXPLICIT Flow Control, is there a clear specified stream control

- Parameter, additional parameter field, parameter can have multiple, each parameter is composed of the following fields:

```
Code, 1byte, identification type
 Length, length
 Corresponding data
```

Then the COPT function package:

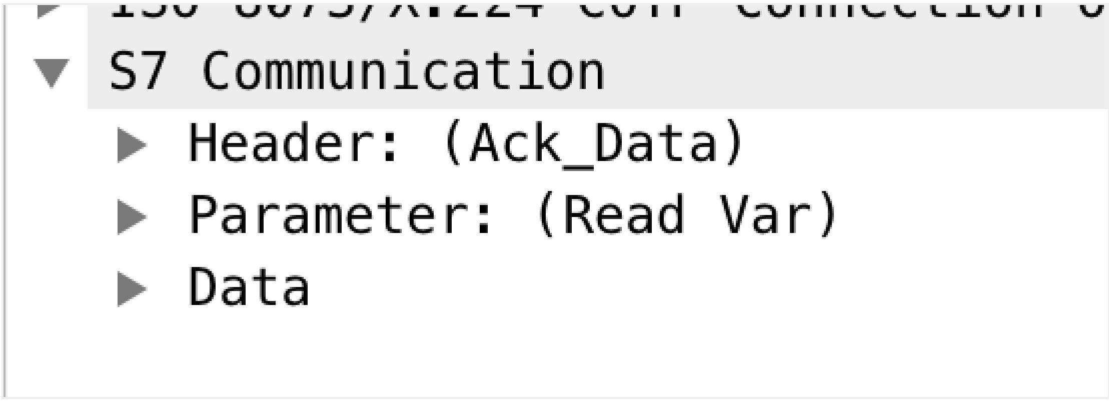
- Length, 1byte, length
- PDU type， 1
  Byte, the picture is 0x0f, which is transmitted for data, and the Type is not used for it, this is no longer mentioned.
- Option, 1byte, partitioned in units:
  The first bit, identifies whether it is the last packet (from this, the COPT protocol will be divided into several unit transfer; the post seven, identify the Number of the TPDU.

S7COMM protocol

I finally came to the last S7COMM protocol, and its structure is very simple, mainly divided into three parts:

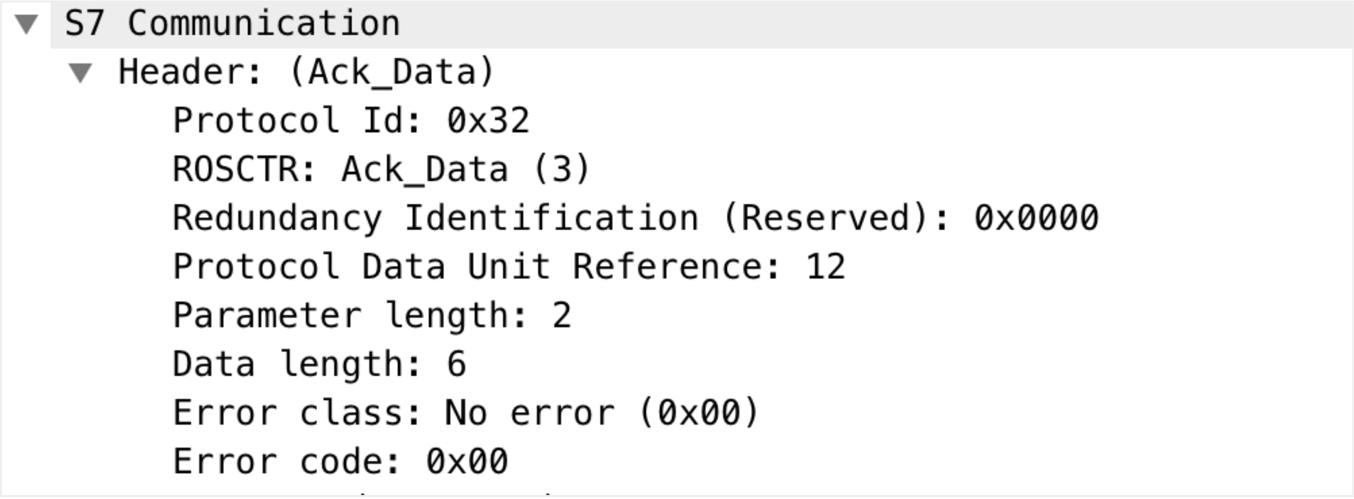
- Header, mainly data descriptive information, the most important thing is to indicate the type of PDU
- Parameter, parameters, as different types of PDUs have different parameters

- Data, specific data



```
▶ ISO 8073/X.224 COTP Connection-O
▼ S7 Communication
  ▶ Header: (Ack_Data)
  ▶ Parameter: (Read Var)
  ▶ Data
```

header：

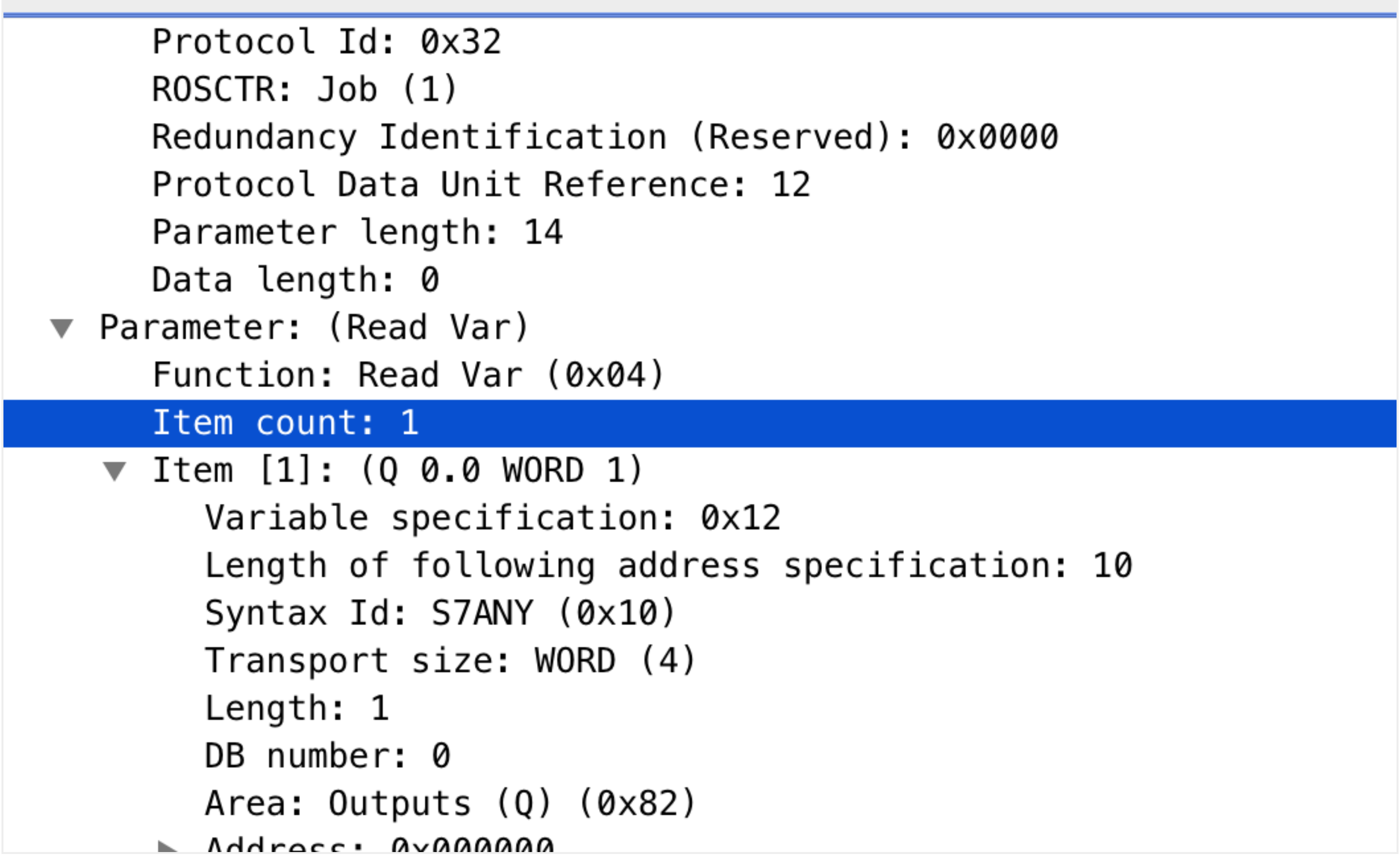


```
▼ S7 Communication
  ▼ Header: (Ack_Data)
      Protocol Id: 0x32
      ROSCTR: Ack_Data (3)
      Redundancy Identification (Reserved): 0x0000
      Protocol Data Unit Reference: 12
      Parameter length: 2
      Data length: 6
      Error class: No error (0x00)
      Error code: 0x00
```

- Protocol ID, 1 byte, the ID of the agreement, 0x32
- Rosctr, 1byte, PDU type, generally by the following:

```
0x01, job, is the meaning of work, the master device issues "work" command through
 0x02, ACK, 0x02, confirmation
 0x03, ACK DATA, from the device's response to the master's JOB
 Reserved, 2byte, reserved
 PDU Reference, PDU Reference
 Parameter Length, the length of parameters
 Error Class, error type, like 0x00 in the picture, is there is no mistake, and the
 Error Code, error code, combined with error type to determine the error, the 0x00
```

Let's see the specific traffic package:

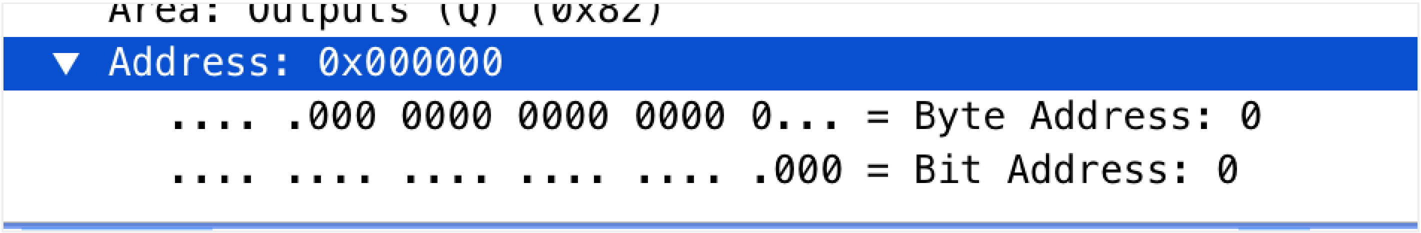


```
      Protocol Id: 0x32
      ROSCTR: Job (1)
      Redundancy Identification (Reserved): 0x0000
      Protocol Data Unit Reference: 12
      Parameter length: 14
      Data length: 0
  ▼ Parameter: (Read Var)
      Function: Read Var (0x04)
      Item count: 1
    ▼ Item [1]: (Q 0.0 WORD 1)
        Variable specification: 0x12
        Length of following address specification: 10
        Syntax Id: S7ANY (0x10)
        Transport size: WORD (4)
        Length: 1
        DB number: 0
        Area: Outputs (Q) (0x82)
      ► Address: 0x000000
```

It can be seen that the PDU is Job, which is the master device in the sending order, and through parameter can be seen, Function is the read of 0x04, which is read data, Item Count means follows several Item, the PDU is one, So 1. And this item structure has to talk about it alone:

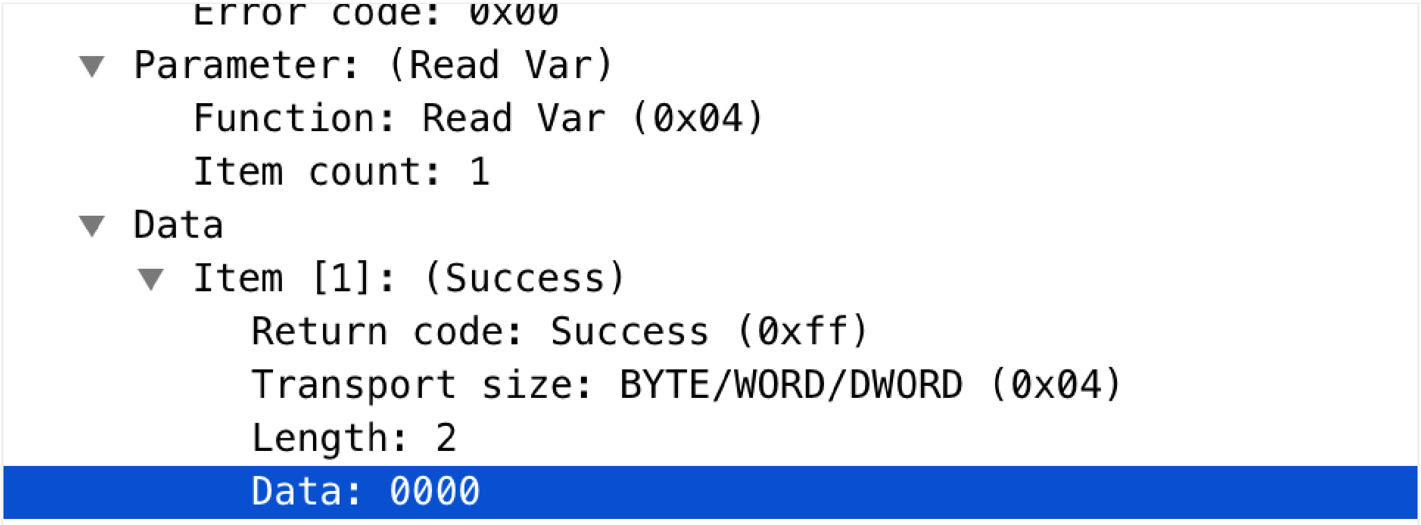
- Variable Specification, 1byte, usually 0x12 (I have never seen anything else ...)
- Length, Length of Following Address Specification, Data Length
- Syntax ID, symbol ID, a flag determines some format issues, here is the meaning of 0x10 is Address Data S7-Any Pointer-Like dbx.dbxx.x, mainly for subsequent addressing
- The transmission size can also be considered to be a transmission type, here is 4, that is, Word
- DB Number is the meaning of the data block number, 0 is not in the data block.
- Area, "things" to be operated, such as 0x82, is the output of the read device, can be seen by this bit, the data we have to read is not in DB, so DB Number is 0, if it is DB, this 1byte should 0x84

- Address, specific address, as shown below, the top five are useless, the sixth to the twenty-first is the BYTE address, the last three is the address of Bit



```
    Area: Outputs (Q) (0x82)
  ▼ Address: 0x000000
      .... .000 0000 0000 0000 0... = Byte Address: 0
      .... .... .... .... .... .000 = Bit Address: 0
```

Let's take a look at the corresponding PDU. Headssencing is not intercepted by Header, Header is most worthy of concern is the type of PDU. Here is 0x03, that is, we have previously mentioned for Job



```
      Error code: 0x00
  ▼ Parameter: (Read Var)
      Function: Read Var (0x04)
      Item count: 1
  ▼ Data
    ▼ Item [1]: (Success)
        Return code: Success (0xff)
        Transport size: BYTE/WORD/DWORD (0x04)
        Length: 2
        Data: 0000
```

The Paramter section can be seen that Function is the same as the Job PDU. The Data section is the specific data of the biography. Return Code is the return code, used to identify the result of Job to make the work, 0xFF, the meaning of success, in addition to this, there are still the following:

- 0x01, hardware error
- 0x03, what you want to access is not allowed to access
- 0x05, the address is bound
- 0x06, your request data type and request "things" data type is inconsistent

Then, the length of DATA (the length of true DATA, does not include front), and finally the specific DATA, it can be seen, here is read 0x0000.

Intelligent Recommendation

2.PCIE protocol analysis

Starting today, we explain the PCIE protocol content analysis section, this chapter is divided into 4 sections content, as follows: (1) Section 1: Preliminaries point We know that to understand FPGA P...

2. Analysis of the MODBUS protocol

0x01 first met Modbus Modbus is a serial communication protocol that is published by Modicon (now Schneider Electric) in 1979 to communicate with a programmable logic controller (PLC). The current MOD...

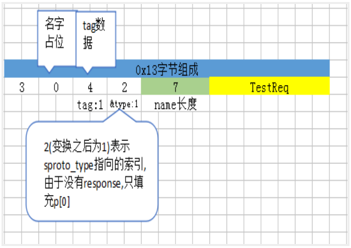
终端启动服务器程序，测试http的请求报文：

```
edu@edu:~/share/dfs$ ./a.out
listen client @port=8000...
GET /html/index.html HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, application/x-ms-appl
ication, application/xaml+xml, application/x-ms-xbap, */*
Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; W
OW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET C
LR 3.0.30729; .NET CLR 3.5.30729)
Accept-Encoding: gzip, deflate
Host: 192.168.31.109:8000
Connection: Keep-Alive
```

http://blog.csdn.net/tennysonsky

Analysis of HTTP protocol (Part 2)

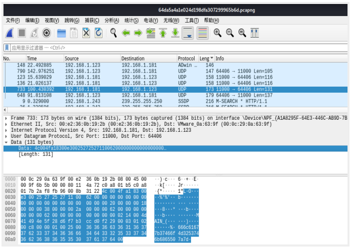
1.1 Server test code Server test code: The browser enters the url address: 1.2 Description of request message format The HTTP request message is composed of four parts: request line, request header, b...



0x13字节组成					
3	0	4	2	7	TestReq
		tag:1	&type:1	name长度	

Analysis of skynet sproto protocol (2)

This article tries to analyze the sproto protocol against the code. A typical sproto protocol looks like this: According to the definition of the code, a complete protocol structure sproto mainly incl...


64da5a4a1e024d198dfa307299965b6d.pcapng
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>

```
No.     Time          Source          Destination       Protocol  Leng  Info
    148 22.492885     192.168.1.123   192.168.1.181     ADwin …   146
    790 142.976251    192.168.1.123   192.168.1.181     UDP       147 64406 → 11000 Len=105
    123 15.639029     192.168.1.181   192.168.1.123     UDP       158 11000 → 64406 Len=116
    136 21.026137     192.168.1.181   192.168.1.123     UDP       158 11000 → 64406 Len=116
    733 100.430392    192.168.1.181   192.168.1.123     UDP       173 11000 → 64406 Len=131
    648 91.813108     192.168.1.123   192.168.1.181     UDP       179 64406 → 11000 Len=137
      9 0.329000      192.168.1.243   239.255.255.250   SSDP      216 M-SEARCH * HTTP/1.1

> Frame 733: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface \Device\NPF_{A1A8295F-64E3-446C-AB9D-7B
> Ethernet II, Src: 00:e2:36:0b:19:2b (00:e2:36:0b:19:2b), Dst: VMware_0a:63:9f (00:0c:29:0a:63:9f)
> Internet Protocol Version 4, Src: 192.168.1.181, Dst: 192.168.1.123
> User Datagram Protocol, Src Port: 11000, Dst Port: 64406
v Data (131 bytes)
    Data: 4c004fa18300e30025272527110062000000000000000000…
    [Length: 131]

0000  00 0c 29 0a 63 9f 00 e2  36 0b 19 2b 08 00 45 00   ··)·c··· 6··+··E·
0010  00 9f 6b 5b 00 00 80 11  4a 72 c0 a8 01 b5 c0 a8   ··k[···· Jr······
0020  01 7b 2a f8 fb 96 00 8b  31 22 4c 00 4f a1 83 00   ·{*····· 1"L·O···
0030  e3 00 25 27 25 27 11 00  62 00 00 00 00 00 00 00   ··%'%'·· b·······
0040  00 00 00 00 00 00 00 00  00 00 00 20 00 00 00 18   ········ ··· ····
0050  00 00 00 38 00 00 00 2a  00 00 00 62 00 00 00 00   ···8···* ···b····
0060  00 00 00 62 00 00 00 00  00 00 00 02 14 00 4d      ···b···· ·······M
0070  41 49 4e 5f 28 d6 f7 b3  cc d0 f2 29 00 03 01 02   AIN_(··· ···)····
0080  00 c8 00 00 01 00 25 00  36 36 36 63 36 31 36 37   ······%· 666c6167
0090  37 62 33 37 34 36 36 66  34 64 33 32 35 33 37 34   7b37466f 4d325374
00a0  36 62 36 38 36 35 35 30  37 61 37 64 00            6b686550 7a7d·
```

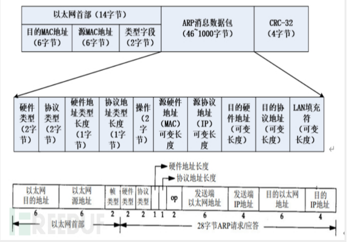
CTF-Industrial Protocol Analysis 2

Industrial Agreement Analysis 2 Source of topic: 2019 Industrial Information Security Skills Competition Individual Online Competition First Session Topic description: During the inspection and evalua...

More Recommendation

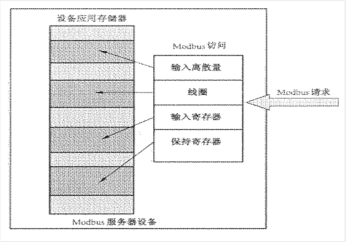
MySQL-Protocol Analysis (Part 2)

MySQL protocol analysis Protocol header Protocol header agreement type Interaction when connected Agreement description handshake packet auth packet ok packet error packet resultset packet resultset p...



Experiment 2 ARP protocol analysis

1. Experimental purpose 1. Analyze the format of the ARP protocol 2. Understand the analysis of the ARP protocol 3. Master the ARP-related command II. Experiment topology Third. Experimental tool GNS3...

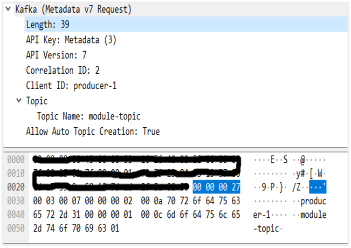


MODBUS protocol analysis (2) - Agreement

1. Protocol description Universal Modbus message frames can be divided into two parts: The 1MODBUS protocol defines a simple protocol data unit (PDU) independent of the infrastructure layer; 2 Specifi...

KCP protocol and source analysis (2)

Original link:KCP protocol and source analysis (2) KCP protocol and source analysis (2) Core function Function IKCP_SEND Function IKCP_RCV Function IKCP_INPUT Function IKCP_FLUSH refer to Core functio...



```
v Kafka (Metadata v7 Request)
    Length: 39
    API Key: Metadata (3)
    API Version: 7
    Correlation ID: 2
    Client ID: producer-1
  v Topic
      Topic Name: module-topic
    Allow Auto Topic Creation: True

0000  [redacted]                                          ····E··S ··@·····
0010                                                      ········ ·y#·[·W·
0020                                 00 00 00 27          ··9·P·}· /Z···'
0030  00 03 00 07 00 00 00 02  00 0a 70 72 6f 64 75 63   ········ ··produc
0040  65 72 2d 31 00 00 00 01  00 0c 6d 6f 64 75 6c 65   er-1···· ··module
0050  2d 74 6f 70 69 63 01                                -topic·
```

Kafka protocol analysis of wireshark 2

Article Directory SASL authentication SaslHandshake handshake negotiation request response protocol analysis SaslAuthenticate request response protocol analysis MedaData protocol analysis SASL authent...

Related Posts

- Analysis of data fields in the S7comm protocol
- S7comm protocol analysis of the data fields (1)
- S7Comm Plus protocol research
- Siemens PLC protocol-S7COMM
- Siemens PLC protocol-S7COMM-extended
- S7Comm Plus protocol research dynamic debugging two
- Dynamic debugging of S7Comm Plus protocol research
- Analysis of Siemens S7comm-plus communication process and replay attack
- 802.11n protocol analysis (2)
- MySQL protocol analysis (2)

Popular Posts

- Keywords: packet
- idea integrates tomcat and solves the console garbled problem
- [SOJ 639] trees
- Xiaobaixue front-end ---------------CSS basic grammar
- MYSQL date and time type format (detailed introduction)
- Network stream (template transfer)

- Layui jq finds the elements of the first element
- Python Algorithm Learning: Competitive Code Programming-Lanqiao Cup School Trial (Preliminary) Replay
- Freeswitch startup service script
- RISCV's cache

Recommended Posts

- Data Structures and Algorithms basis
- Front End Knowledge Sharing - SHEETJS Usage Experience
- Solve Endnote's reference to the problem without GBT7714 format
- Day04 (array)
- Mac installation git

- Image and large array types
- Convert grayscale image to pseudo-color image-pseudo-color processing
- GHOST blog build
- Android application component - Service
- Krpano tutorial - view tag Chinese description

Related Tags

Siemens PLC Code decompile s7comm

Industrial Control Protocol Series

Safety

plc

S7

socket

java

Internet of Things

The internet

Industrial control safety knowledge