S7Comm Plus protocol research

tags: Industrial Control Protocol Series Safety

1 Overview

Recently, a new version of Siemens S7-1200PLC has been started, the firmware version is V4.2.3, the communication protocol is S7comm-Plus, and it has fully supported the authentication and data encryption of the communication process. In fact, after the PLC worm was proposed in April 2016, the S7comm-Plus protocol has been fully enabled in V4.0 and later firmware versions, and the security has been greatly improved. Simple and rude replay attacks are no longer so easy. It worked. At the blackhat conference in August 2019, Israeli researchers successfully developed a fake workstation that simulates TIA Portal, which can successfully interact with the new version of Siemens PLC (S7-1200, S7-1500), and perform start/stop and logic tampering. For various operations, this seems to mean that PLC worms can also be implemented in higher versions of PLCs.

To achieve an attack on the new version of the PLC, it is necessary to study the S7comm-Plus protocol, understand the entire communication handshake, and the authentication and encryption process. The domestic NSFOCUS and Qiming have conducted reverse analysis of the core communication DLL file to achieve the PLC Start/stop attack. After studying the relevant papers of blackhat2017 and 2019, I found that to understand the whole process, there is still a bit of cryptography foundation. Through many experiments and understanding of the paper, I have some personal understanding of the S7comm-Plus protocol. This article is formed for reference Exchange and study, hope to correct the shortcomings to promote common progress.

2. Environment configuration

The basic environment configuration of the entire experimental study is as follows: Win7x64 virtual machine,

PLC： S7-1200, 6ES7 212-1BE40-0X0B

Firmware: V4.2.3

Software： TIA Portal V14

S7Comm-Plus Wireshark dissector plugin: V0.0.8

In the article, the protocol is divided into three versions: P2-, P2, P3, etc. Different versions of TIA Portal software are combined with different PLCs, and different protocol versions are used for communication:

According to this classification method, the P2-version protocol is used in my research environment, but the communication data is obviously the P3 version protocol:

After verifying with the author and Siemens, the new version of S7-1200PLC uses the P3 version protocol. In addition, four versions of TIA Portal V13, V14, V14SP1, V15 are configured and compared, and the following basic conclusions are drawn: 1. S7-1200 PLC with firmware version V4.2 and above must use V14 and above TIA Portal for configuration (V13 only supports V4.1 firmware); 2. V13 supports both 32-bit and 64-bit operating systems, V14 and above only support 64-bit systems; 3. V14 is not Win10 system; 4. V14 The core communication DLL (OMSp_core_managed.dll) of the V14SP1 version is the same. The version of V15 has been updated. The detailed information of this DLL corresponding to the three versions of V13, V14 and V15 is as follows:

Therefore, the appropriate TIA version can be selected for installation according to the target PLC firmware version.

3. Analysis of communication process

3.1. Handshake process

The TCP/IP implementation of the S7Comm-plus protocol relies on the block-oriented ISO transmission service, and its OSI model is as follows:

Inheriting the "session id" introduced in the previous version to prevent replay attacks, the new version of the protocol introduces key protection and data encryption mechanisms, and encrypts every data packet with operational functions. This makes it more effective to deal with attacks such as replay attacks, man-in-the-middle attacks, and session hijacking. The basic process of communication and interaction between the upper computer (TIA) and PLC is shown in the following figure:

1. Handshake initiation: communication handshake initialization, that is, the CR/CC data packet part;

2. Challenge: TIA and PLC establish S7Comm-Plus Connection, PLC side generates 20-byte random number, and feeds it back to TIA side;

3. StructSecurityKey: TIA establishes S7Comm-Plus Connection with PLC, and TIA generates authentication data based on the random number (Challenge) combined with the public key;

4. ACK: TIA establishes S7Comm-Plus Connection with PLC, PLC side uses private key to decrypt authentication data (StructSecurityKey), after authentication is successful, reply TIA side OK, communication establishment is successful;

5. Function: TIA terminal sends data with functional operation to PLC (such as PLC start/stop);

In the data interaction of each of the above steps, the same "session id" is used to capture the communication process. The specific data is as follows:

Further analysis of the authentication process in the communication establishment process (ie S7Comm-Plus connection):

1. TIA sends M1 to the PLC to start the session, and uses the "CreateObject" function code to create the "ClassServerSession" object:

2. The PLC responds to TIA's request and replies to M2. M2 contains the PLC firmware version and the 20-byte random number ServerSessionChallenge, as well as the session id:

3. After TIA receives M2, according to the random number (a 20-byte random number, only the middle 16 bytes are selected in the actual calculation process, and the first two are not involved in the actual calculation) and combined with the public key. Use complex encryption algorithms (including basic XOR, Hash such as SHA-256, MACs such as HMAC-SHA-256, CBC-MAC, AES-CTR mode, AES-ECB mode, ECC) to generate authentication data and respond to PLC And reply to M3, the key part of the authentication data that needs to be paid attention to is the structure of "StructSecurityKey", in which the 180-byte "SecurityKeyEncryptedKey" is the most important field:

4. After the PLC receives the M3, it uses the private key to decrypt and authenticate the encrypted data. If the authentication succeeds, it will reply to the TIA with an M4 packet (a packet with a length of 86 in the figure above).

5. After the authentication is successful, TIA sends a data packet with functional operations to the PLC. TIA uses a private algorithm (using a session key) to calculate the content of the data packet to obtain a 32-byte IntergrityPart field, and the PLC receives the function code data packet After that, first verify the IntergrityPart field, and execute the corresponding function code action if the verification is passed.

3.2, encrypted fields

It can be seen from the basic communication process that the generation of the encrypted field in the M3 data packet is the key to successfully establishing communication with the PLC, and the subsequent calculation of the IntergrityPart is the key to successfully operating the PLC. The process of data encryption is a complex cryptographic algorithm practice process. As can be seen from the above figure, the Wireshark plugin has correctly interpreted most of the fields in the data, but a few fields, such as "SecurityKeyEncryptedKey" in "StructSecurityKey" The field has not been fully identified. Combining the article and the actual data packet, we will specifically identify the 180-byte "SecurityKeyEncryptedKey".

The structure identified by Wireshark and the decomposition of the field in the article are shown in the following figure:

Further decomposition and correspondence of actual data:

```
0000ad de e1 feb4 00 00 00 01 00 00 00 01 00 00 00

0010d1 58 ff a4 13 13 c0 7b 01 01 00 00 00 00 00 00

00201a 73 08 1f 09 6b 42 bd 10 01 00 00 00 00 00 00

0030a7 c0 65 16 c5 af f4 ff c6 b8 cb 5d b3 35 3d 44

00404d 48 3b 5da5 48 81 cc 82 85 fd 1a f5 5d 3e 3c

005086 c5 6f ae 5d 59 cb bee6 c9 99 fa 39 f6 3d ac

00603a 12 a5 4d 93 b1 f3 8d c7 46 7a f973 86 90 c0

0070fd 56 f2 ea 4f 7d 7b d7 1d 67 f0 1aa9 9b 46 89

008030 5b d1 bf e8 7e c5 b2 96 5e 55 cd72 4a 96 cc
```

```
0090e1 5a 1a 0e 9d 79 12 4f a4 46 f9 0e 4b d7 05 a7

00a0cc 4a a4 3f61 0c c3 b5 d5 bd dd 70 b2 be f0 be

00b0e2 a9 93 ca
```

Magic byte:0xFEE1DEAD is magic, 4 bytes are fixed, and it is in little endian mode;

Length: The length of the field, which is 180;

Symmetric key checksum: 8 bytes, which is the KDK ID Header mentioned in the article;

Public key checksum: 8 bytes, that is, the Public key ID Header mentioned in the article;

The 8 bytes are the same for all S7-1200PLCs of the same model and firmware, because Siemens uses the same public key for PLCs of the same model and the same model, and the checksum here is the public key to calculate the SHA- 256 and take the first 8 bytes. In the experiment, the public key of S7-1200PLC can be obtained through dynamic debugging, as shown in the following figure: 40 bytes:

The article also pointed out that the PLC's public key information is stored in the TIA installation directory: Siemens/Automation/Portal V14/Data/Hwcn/Custom/Keys), but its files are stored in an encrypted manner.

EG1: 20 bytes;

EG2: 20 bytes;

In the communication process, TIA randomly generates 20 bytes as PreKey, uses elliptic curve encryption algorithm and PLC's public key to encrypt PreKey, and waits until the content is the content of EG1 and EG2;

Kxv3: 20 bytes, the content here is not fully understood, that is, the Nonce marked in the article;

IV: 16 bytes, the initial data encrypted in AES Counter mode;

Encrypted Challenge: 16 bytes, this content is the result of using AES-CTR mode to encrypt 16 bytes of the 20-byte random number in M2, together with the IV part, namely: AES-CTR (Challenge, KEK ,IV)

Encrypted KDK: 24 bytes, calculated using AES-CTR (Challenge, KDK, IV);

Encrypted Checksum: 16 bytes, using AES-ECB (Checksum, ECK), where Checksum=TB-HASH (CS, Encrypted KDK, Encrypted Challenge)

After one-to-one correspondence and identification of each encrypted data field, looking back at the entire key generation algorithm and exchange process, the idea slowly becomes a little clearer:

Sort out the key points in the whole process:

1. Identify the input data: The most obvious input data is the PLC Public key, which can be obtained directly, and the PreKey, which is a 20-byte random number randomly generated by TIA, which can be grabbed from the memory during dynamic debugging ; Challenge, can be obtained from M2 data;

2. Various complex cryptographic algorithms: Regarding the elliptic curve-like encryption algorithm, a fixed 40-byte key is used, representing a 160-bit elliptic curve point. The base point G of the elliptic curve is hard-coded in the

OMSp_core_managed.dll file In the process of dynamic debugging, it can also be grabbed from the memory. The G of S7-1500 and S7-1200 are as follows:

4. Summary

Based on the preliminary analysis of Siemens' latest S7Comm-Plus communication protocol, the whole process uses very complex authentication and encryption methods, and it is not a simple matter to crack and bypass. However, because the authentication is unilateral in the communication process, that is, TIA has authenticated the PLC, but the PLC has not authenticated the TIA, so it can forge the TIA to establish communication with the PLC; in addition, the PLC of the corresponding type and firmware version uses the same The private-public key pair means that a successful attack on one S7-1200 is completed, that is, an attack on all S7-1200 is realized.

Although the authentication and encryption in the communication process is extremely complicated, the article has realized the attack on the new version of S7-1500PLC. The attack ideas are summarized as follows:

1. Capture the input data required for encryption authentication through dynamic debugging, such as the random number generated by TIA and the Challenge returned by M2;

2. Locate the corresponding encryption function;

3. Use the encryption function to calculate the input data to get the correct data packet field, combine the fields into a complete data packet and send it to the PLC for verification. (There are two ideas here: ① The article pointed out that the Ctypes module of Python is used to wrap the core communication DLL:

OMSp_core_managed.dll to construct the correct input parameters to realize the entire encryption authentication process; ② The encryption process is dynamically debugged to clarify the encryption authentication process , Locate the relevant function function, and at the same time extract the function function with the reverse analysis, the input parameter of the constructor function, and complete the entire encryption authentication process.)

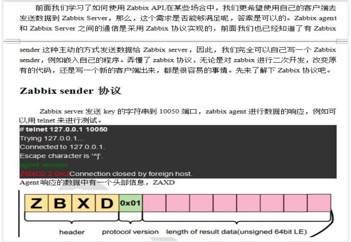
Reference materials:

[1] https://i.blackhat.com/USA-19/Thursday/us-19-Bitan-Rogue7-Rogue-Engineering-Station-Attacks-On-S7-Simatic-PLCs-wp.pdf

[2] https://i.blackhat.com/USA-19/Thursday/us-19-Bitan-Rogue7-Rogue-Engineering-Station-Attacks-On-S7-Simatic-PLCs.pdf

Intelligent Recommendation

Research on zabbix sender protocol



前面我们学习了如何使用 Zabbix API,在某些场合中，我们更希望使用自己的客户端去发送数据到 Zabbix Server，那么，这个需求是否能够满足呢，答案是可以的。Zabbix agent 和 Zabbix Server 之间的通信是采用 Zabbix 协议实现的，前面我们也已经知道了有 Zabbix sender 这种主动的方式发送数据给 Zabbix server，因此，我们完全可以自己写一个 Zabbix sender，例如嵌入自己的程序。弄懂了 zabbix 协议，无论是对 zabbix 进行二次开发，改变原有的代码，还是写一个新的客户端出来，都是很容易的事情。先来了解下 Zabbix 协议吧。

Zabbix sender 协议

Zabbix server 发送 key 的字符串到 10050 端口，zabbix agent 进行数据的响应，例如可以用 telnet 来进行测试。

```
# telnet 127.0.0.1 10050
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
agent.version
ZBXD2.2.0rc2Connection closed by foreign host.
```

Agent 响应的数据中有一个头部信息，ZAXD

Research on zabbix sender protocol time2013-12-22 Authoritnihao Mailbox Bloghttp://www.itnihao.com For quotation, please indicate the above information, thank you for your cooperation Prerequisites fo...

Protocol Research Subtotal

Compile protobuf Native compilation steps Use protobuf-gradle-plugin Avoid executing every command line Added protobuf-gradle-plugin to project build.gradle Module build.gradle configuration Configura...

发布消息

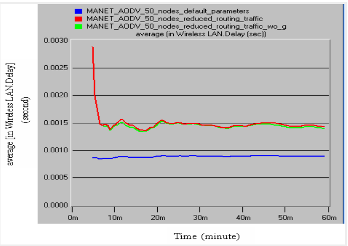
```
@Test
public void testPublish() {
    try {
        MqttClient client = new MqttClient("tcp://192.168.218.129:1883","pahotest");
        client.connect();
        MqttMessage msg = new MqttMessage();
        msg.setPayload("hello world!".getBytes());
        client.publish("mqttdemo/test", msg);
        client.disconnect();
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

订阅消息

```
@Test
public void testSubscribe() {
    try {
        MqttClient client = new MqttClient("tcp://192.168.218.129:1883","pahotest");
        MqttConnectOptions options = new MqttConnectOptions();
        options.setCleanSession(true);
        options.setKeepAliveInterval(30);
        client.connect(options);
        client.subscribe("mqttdemo/test", 2);
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

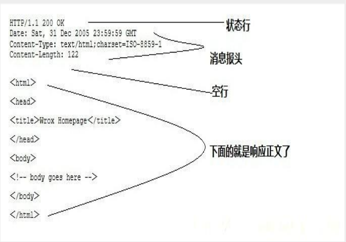
Research on MQTT protocol

The MQTT protocol is a lightweight Ethernet data exchange protocol, which is more and more widely used in the Internet of Things. This article briefly introduces the relevant knowledge points of the M...



Simulation Research of AODV Protocol

Mobile Ad Hoc Network (MANET) is the mobile packet wireless network called in the early military research. It is a collection of interconnected radio stations, computer hardware and software based on ...



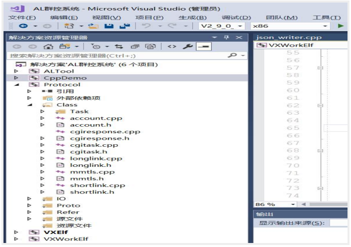
```
HTTP/1.1 200 OK
Date: Sat, 31 Dec 2005 23:59:59 GMT
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>
```

Research on HTTP Protocol

introduction Detailed HTTP protocol URL articles Detailed HTTP protocol request Detailed HTTP protocol response chapter

HTTP protocol detailed message header HTTP status code HTTP request method How H...

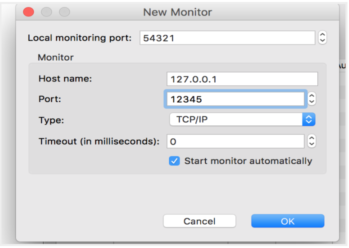
More Recommendation



```
解决方案'AL群控系统' (6 个项目)
  ALTool
  CppDemo
  Protocol
    引用
    外部依赖项
    Class
      Task
      account.cpp
      account.h
      cgiresponse.cpp
      cgiresponse.h
      cgitask.cpp
      cgitask.h
      longlink.cpp
      longlink.h
      mmtls.cpp
      mmtls.h
      shortlink.cpp
      shortlink.h
    IO
    Proto
    Refer
    源文件
    资源文件
  VXElf
  VXWorkElf
```

Research on PC WeChat Protocol

The MMTLS used by the encryption layer of the PCWX protocol, the private long chain of the upper protocol and the short HTTP chain, have now all been implemented You can log in successfully, an...



Local monitoring port: 54321

Host name: 127.0.0.1

Port: 12345

Type: TCP/IP

Timeout (in milliseconds): 0

☑ Start monitor automatically

Research on soap protocol

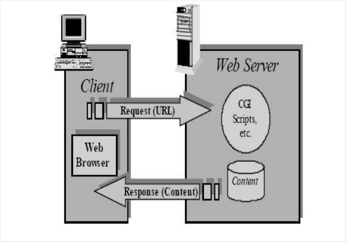
1. Turn on tcp agent monitoring Here we use eclipse proxy monitoring window->show view->others select TCP/IP Monitor Select Properties in the TCP/IP Monitor view or find TCP/IP Monitor in the pr...

Research on AirPlay Protocol

The AirPlay protocol implements the transmission of software layers for Apple devices to display a set of private protocols in the information delivery group. This technical support automatically disc...

Onfi protocol research

Overview ONFIIs a set of public standards in NAND Flash Official website:http://www.onfi.org/ The chance is coincidental, just look Official website has PDF download, latest version 5.0, 383 Going up ...



WAP protocol research notes WAP transmission protocol

Share the artificial intelligence tutorial of my teacher, God! Zero basis, easy to understand! You are also welcome to reprint this article. Sharing knowledge, benefiting the people and realizing the ...

Related Posts

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

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

Industrial Control Protocol Series

Safety

plc

S7

socket

java

Industrial control

Internet of Things

The internet

Siemens PLC Code decompile s7comm