Research on Industrial Control Network Security Data Acquisition System

Qiucheng Xia*, Peng Wu*
CSIC Information Technology Co., Ltd., Lianyungang, China

*Corresponding author e-mail: 2247325194@qq.com, *dubyrd@163.com

Abstract. In this paper, based on the characteristics of industrial control network security data, research data analysis and data storage technology, design and development of industrial control network security data acquisition system, and through the AFL fuzz method for fuzzy testing, improve the system's robustness and endogenous security performance. The realization of massive heterogeneous data collection and efficient normalization of industrial control networks can better serve the data layer and promote the rapid development of digitalization of industrial enterprises.

1. Introduction
The industrial enterprise safety operation center plays an important role in the enterprise safety protection and is the basis for the enterprise's safe operation and production. Through the combination of active and passive methods, the real-time and uninterrupted collection of massive heterogeneous log information in security equipment, network equipment, host computers, operating systems, and various application systems (such as PLC, DCS, SCADA system) from various manufacturers in the industrial control network, and collect this information in the audit center for centralized storage, backup, query, audit, alarm, response, and issue rich report reports to learn about the overall security operation situation of the entire network and achieve a full life cycle log management. It can be seen that the authenticity, real-time and integrity of the data collected by the industrial control network are critical to the security operation center.

2. Characteristics of collected data
The main characteristics of data collection are as follows:
(1) Multiple data receiving methods: TCP, UDP, FTP, SFTP;
(2) Support various formats of security logs: XML, JSON, CSV, SYSLOG, NETFLOW, SNMP, etc.;
(3) Pre-process security logs, generate standardized formats, and enter data warehouse;
(4) Support one data source corresponding to multiple libraries, and one library corresponding to multiple data sources;
(5) Configuration management function, through the interface, configure data sources, libraries, tables, security logs, rules, establish data flow direction, define which data source the data comes from, and which table in which library;
(6) Support asset (network equipment, such as host computer, industrial security gateway, industrial control security audit, host security guard, switch, router, intrusion detection, etc.) operation management log, the format is shown in the following table.
Example of log format of industrial control intrusion detection system:
<Type = 0> <space> <content>

| Alarm level | Urgent (0) |
|-------------|------------|
| Content     | Event description <space> source IP address <space> source port <space> destination IP address <space> destination port |
| Example     | <0> 2019-03-12 20:12:23 kemel IDS 0 DistCC daemon command execution vulnerability 192.168.10.244 138 192.168.10.255 138 |

### Table 1. Intrusion protection event table.

3. Data collection process
Collect and efficient normalization massive heterogeneous data of resource layer objects through various protocols such as SYSLOG, SNMP Trap, FTP, SFTP, JDBC, ODBC, and Net flow, and transfer the data to the data layer. The main process is shown in Figure 1.

![Flow chart of data collection.](image)

**Figure 1.** Flow chart of data collection.

4. Design of data acquisition system
It can be seen from the data collection process that data enters the collector from various objects through various protocols. On the one hand, the collector normalization and transmits data to the buffer queue of the management center (cluster); On the other hand, it normalizes the log event engine, generates events, and transmits the events to the cache queue. The main function of the collector is to support various access protocols, log normalization, and event generation. Its function is shown in Figure 2.
Collector function:

1. The client encapsulates the message according to the message format of the A-end server, and then sends the message to the server;

2. The device log guarantees the integrity of data transmission and reception through TLS, and supports TCP, UDP, FTP, and SFTP;

3. Unspecified device logs are sent to the server through UDP, FTP, SFTP; if you need to send through TCP, you need to develop the corresponding decoder plug-in on the server;

4. For the server, receive the data sent by the client, then unpack according to the decoder plug-in, and forward it to the corresponding output location according to the output configuration of the output plug-in, such as Kafka, Local directory, etc.

4.1. Server principle

The principle of the server is shown in Figure 3.

---

**Figure 2.** Collector function view.

**Figure 3.** The server technology realization diagram of collector.
The detailed description of the server interface is shown in Table 2.

### Table 2. The server interface table of collector.

| Module       | Interface                | Description                                      |
|--------------|--------------------------|--------------------------------------------------|
| Sender       | TCP client               | TCP client                                       |
|              | [TcpInput_5565]          | Type: plugin type                                |
|              | type = "TcpInput"        | Address: listening address                        |
|              | address = "0.0.0.0:5565" | Decoder= protobufDecoder                        |
|              | [KafkaOutput_5566]       | Type: plugin type                                |
|              | Type="KafkaOutput"      | Address: listening address                        |
|              | address = "10.67.10.2:5566" | Message_matcher: data filtering rules |
|              | message_matcher = "TRUE" | startServer()                                    |
| Recver       | GetMesg()                | Application layer receives data                   |
|              | recvMessage              | Receive data from HEKA                            |

### 4.2. Client principle

The principle of the client is shown in Figure 4.

![Client principle diagram](image)

*Figure 4. Client technology implementation diagram of the collector.*

The detailed description of the client interface of the collector is shown in Table 3.

### Table 3. Client interface table of the collector.

| Module       | Interface        | Description                                      |
|--------------|------------------|--------------------------------------------------|
| Sender       | startClient()    | Open the client                                  |
|              | SendMesg()       | Application layer sending data                   |
|              | packMessage      | Encapsulate data into HEKA message type           |
|              | sendMessage      | HEKA Send data to HEKA                           |

### 5. The implementation of data analysis technology

#### 5.1. Data analysis technology

The implementation of data analysis technology is shown in Figure 5.
The data analysis function is as follows:
1) Read data from Kafka;
2) Load the analysis plug-in to analysis the data;
3) Make log enhancement to the parsed data and generate formatted data;
4) Output the formatted data to the Kafka temporary queue for storage, alarm and other services.

5.2. Data storage technology
The data storage technology implementation is shown in Figure 6.

The data storage function is as follows:
1) Read the logs generated by analysis from the Kafka temporary queue;
2) Generate data to Hive warehouse;
(3) Generate data to Elasticsearch.

6. System test
AFL fuzz is a fuzzing tool. It belongs to a fuzzing tool. It encapsulates a GCC / CLang compiler, which is used for instrumentation in the process of recompiling the code under test. After the instrumentation is completed, AFL fuzz can input different parameters for the compiled code, track the execution path of the tested code, and determine whether the input mutation can trigger a new known or unknown execution path. The implementation path is shown below.

1. The Main function performs initialization and option processing first;
2. Execute all testcases (perform_dry_run) prepared in advance in the input folder to generate the initial queue and bitmap;
3. Selection queue through cull_queue to reduce the amount of input;
4. Then carry on the fuzz and while (1) loop.

By testing the data acquisition system, the results are as follows:

![Data acquisition system test results.](image)

Through the fuzzy test tool test, analyze and improve the cracks in the system to improve the system's robustness and endogenous safety performance.

7. Conclusion
The system developed according to the requirements of industrial control network security data collection, after fuzzy testing, can carry out mass heterogeneous data collection and efficient normalization of industrial control networks, better serve the data layer, and promote the rapid development of digitalization of industrial enterprises.

References
[1] Bin Guo, Meng Lei, Deyi Wang, Design of Situation Awareness Data Collection in Industrial Control Network, Automation Panorama. 2019 Issue S2, pp 110-113.
[2] Baocheng Xie, Yugang Han, Industrial control calculation processing and data information collection equipment, China. Patent CN201711244965.5. (2018).
[3] Wenbo Du, Xiaoguang Du, Tianjie Wen, Information security monitoring device and industrial control system, China. Patent CN201820397216.X. (2018).
[4] Yi Wang, Xiufeng Wang, Lingyu Fan, An Industrial Big Data Acquisition Network System, China. Patent CN201811618625.9. (2019).

[5] Tao Fu, Industrial control network security situation awareness system, China. Patent CN201811637918.1. (2019).