Research on Analytical Method of Private Power Protocol

Jie Wang, Zhiming Zhong, Zhongfei Wu, and Guanke Liu
Dongguan electric power supply bureau, Dongguan 523000, China

Abstract. Based on the study of protocol reverse engineering, this paper proposes a method of deep message parsing for private power protocols. It can parse industrial control protocols, extract protocol fields and state machines without knowing protocol specifications. This method can be used to monitor the flow of private protocol network safely, and it is of great significance to ensure the safe and stable operation of power grid.

1. Introduction
In the process of network traffic security monitoring and analysis of power monitoring system, it is necessary to deeply analyze the industrial control protocol message. Besides the well-known protocols such as IEC60870-5, IEC61850 and DNP3, a large number of private protocols are widely used in the power grid, such as IEC104 of the national network, IEC103 extended by various equipment manufacturers, etc. For the parsing of these private protocols, the protocol specifications are not transparent, and the traditional field-by-field parsing method based on protocol specifications can not be used.

On the basis of researching the major methods of protocol reverse engineering at home and abroad, this paper proposes an automated deep message parsing method for private protocols under unknown protocol specifications, and extracts protocol fields and state machines. This method will ensure the security of power grid fully stable operation is of great significance if applied to the security monitoring of private protocol network traffic.

2. Protocol reverse engineering
Protocol reverse engineering refers to extracting protocol grammar, semantics and synchronization information by monitoring and analyzing network input and output, system behavior and instruction execution process of protocol entities without relying on protocol description. In the field of protocol reverse engineering, more in-depth research has been carried out at home and abroad. Considering that the power operation environment is relatively closed and the executable code of the core equipment is difficult to derive, this paper adopts the analysis method based on network traffic as the basic technical route of protocol reverse analysis.

The analysis method based on network traffic takes the captured network message sequence as the analysis object. It is relatively easy to obtain input, flexible to implement, and has strong versatility, which meets the automatic reverse analysis requirements of power industry control protocol.

3. Analytical method of private power protocol
3.1. General scheme
In order to realize the automatic parsing of private protocols, the following technical routes are proposed: firstly, the private protocol traffic is imported into the test system through bypass flow
mirroring, such as splitting, shunting, switch port mirroring, etc. Packet samples are clustered initially in the protocol reverse module by using partial sequence alignment algorithm and unweighted pairwise average algorithm (UPGMA); combined with the obvious structured characteristics of industrial control protocol, the variable domain and invariant domain of message sequence are separated by using multi-sequence alignment algorithm, and the structure information of protocol message is further analyzed. The special semantic field (such as length fields, check field, etc) of message is searched by a self-defined search algorithm, and the protocol format letter is parsed. Information: Using the protocol message structure and semantic information, we get the state transition sequence in each session, construct the state prefix tree, merge the redundant states, and then optimize the state tree to obtain the final minimum deterministic state machine.

3.2. Protocol reverse method

Protocol inversion is mainly to obtain protocol message structure, semantic information, and state information and context relations.

This scheme adopts the following methods to realize protocol reverse analysis:

- Using Multi-Sequence alignment algorithms to obtain message structure information;
- Aiming at common semantic fields in industrial control protocols, heuristic algorithm is set up to match and extract semantic information;
- Protocol state machine inference (including context relations) based on state annotation;
- Inference of Protocol Packet Structure Using Multiple Sequence Alignment Algorithms

Sequence alignment, also known as sequence alignment, aligns two or more sequences together to find out the maximum similarity matching between sequences and identify the differences between them. Sequence alignment methods are often used to study homologous sequences. Especially for biological sequences such as protein sequences or DNA sequences, protocol formats are regarded as languages, and protocol messages have evolutionary similarities. Therefore, sequence alignment can also be used to analyze protocol formats.

The common multi-sequence alignment algorithms are mainly divided into three categories: precise alignment algorithm, iterative alignment algorithm and progressive alignment algorithm. The progressive alignment algorithm adopts greedy idea and iteratively executes the dynamic programming algorithm of double-sequence alignment. It begins with the alignment of two sequences, and gradually adds new sequences until all sequences are added. So far, compared with the other two algorithms, the asymptotic comparison algorithm has obvious advantages in efficiency. Although the result of asymptotic alignment can not be guaranteed to be optimal, the asymptotic alignment can still achieve satisfactory results when the similarity of sequence samples is high.

Therefore, this scheme aims to extract protocol format by progressive multiple sequence alignment, which includes three steps:

1. Computing distance matrix. The most commonly used Smith-Waterman algorithm is used to find out the best local comparison between two samples, and then calculate the similarity between samples, and construct the distance matrix D of the sample set.

2. Constructing and segmenting guided trees.

Unweighted Pair-Group Method with Arithmetic (UPGMA) is used to calculate the distance between subclasses, and the subclasses with the smallest distance are merged step by step. The distance between the subclass C_i and C_j can be calculated by the following formula:

\[ d_{ij} = \frac{1}{|C_i \cup C_j|} \sum_{p \in C_i, q \in C_j} D_{pq} \]

Since there may be many format types in the protocol, if the alignment is forced in asymptotic multiple sequence alignment, a large number of invalid filling bits may be added to the sample. In order to improve the accuracy of sequence alignment, a distance threshold is set. When the DJ is too large, the merging is stopped, and finally multiple guide trees are segmented. In the boot tree, the leaf
node represents the original sample sequence, and the intermediate node represents the alignment sequence obtained by the double sequence alignment of the sub-nodes.

Perform asymptotic multiple sequence alignment. After sequential traversal of the guide tree, double sequence dynamic programming alignment is carried out by using the Needleman-Wunsch algorithm, and the unaligned bytes are filled. When multiple guide trees are constructed, progressive multiple sequence alignment will result in multiple subsets of samples. Each sample subset is analyzed and processed to obtain the protocol structure information.

Heuristic semantic extraction algorithms

Semantic inference based on network traffic is to infer the semantics of a byte (segment) in the protocol according to the value and change characteristics of a byte (segment) in the sample. The semantics of all bytes (segments) in the sample form the protocol format of the sample set. In order to describe conveniently, a structure with specific semantics in the sample structure is called a key field, which is called field for short. According to the static characteristics of the sample, it can identify the interval field, serial field, data field, length field and format identification (FD, Format Distinguish) field. In addition, for structures that do not recognize semantics, they can be identified as unknown fixed fields or unknown variable fields. For each field recognition, a set of semantic inference strategies will be developed. Length field and format identification field are the two most basic fields. This paper mainly introduces the recognition strategies of these two fields.

Length field

The feature of a length field is that its own length is generally 1-4 bytes, and the value is equal to the length of a segment or a continuous segment of the sample, and the scope will not precede the field (but may be in the scope). Its recognition strategy is to determine whether the value of the field is equal to the length of a subsequent segment or a continuous segment, and if it is equal, it will be determined as a length field. Since our semantics inference is carried out within the segment, the scope of the length field may also include the segment. In order to increase the time complexity of the algorithm, a simplification is made without affecting the validity and correctness of the algorithm. The segment after the current segment is taken as a whole, that is, the scope may include either the segment or the segment. All subsequent segments are either excluded. Similarly, due to the role of spacers, in order to reduce the time complexity of the algorithm and error recognition, the text field has merged the samples according to spacers, that is, the samples have been segmented.

Format Distinguish(FD) field

The format identification field is characterized by a small rate of change in its own values, and its values are closely related to the format sequence (classification) that follows (a value corresponds to the current subclass). Discoverer is a format location and recognition algorithm developed by the Cui team of Microsoft Research Institute. Its recognition strategy is to scan from left to right. First, according to semantic inference, some semantic fields are preliminarily judged, and the first FD field is determined. Then, according to the value of FD field, the sub-segments are classified, and then semantic inference and the next FD are made. Fields are identified and format fusion is carried out at last. The advantage of its recognition strategy is that it can locate the FD field more accurately, and the FD field is the field that decides the format sequence after the FD field. So the precondition of identifying the FD field is to identify the other fields after the FD field first, while Discoverer identifies the previous FD field first, and enters according to the value of the FD field. Line classification, semantic inference of subclasses and identification of the next FD field. Thus, the disadvantage of Discoverer is that it reverses the recognition order, and the accuracy of the FD field is low, even the FD field cannot be recognized. In this paper, Discoverer's FD field recognition strategy is improved based on the principle that only one FD field can correspond to one format sequence. The improved strategy is as follows: algorithm 1:

Algorithm 1. An algorithm to infer the FD keyword of the current section

Input: InitialCluster
Output: FDCluster
FDInfer (InitialCluster)
1. UC ← FindNextUC (InitialCluster)
2. SubClusters ← SubClustering (UC, InitialCluster)
3. for each SubCluster in SubClusters
   4. FDInfer (SubCluster)
   5. FormatSeqs ← ExtractFormatSequence (SubCluster)
   6. if (CompareFormatSequence (FormatSeqs) = true)
      7. UC ← FD
      8. UpdateFormat (SubCluster, FD)
      9. FDCluster.Push (SubCluster)
   10. else
      11. return InitialCluster
   12. end if
5. end for
6. return FDCluster

The input InitialCluster is a sample set that has been inferred semantically other than the FD field. The output of the algorithm is the sample set identified by FD field. After inferring the semantics of the current segment except FD, the algorithm first recursively locates the possible location of the FD field from left to right. Since other semantic inferences are based on sequence alignment results of the same type of message in the current segment, the format sequence of the same type of message in the current segment is the same, so the FD field may only be an unrecognized fixed field aligned between different types of messages from left to right, which is designated as an undetermined field (line 1 of the algorithm).

According to the value of the field to be determined, classify it (line 2 of the algorithm), and then call the algorithm recursively to each subclass to continue to locate the FD field backwards (line 4 of the algorithm). After positioning, it confirms whether the undetermined field is an FD field from right to left. The rule of confirmation is to scan backwards from the undetermined field until the next FD field or the last field of the current segment (if there is no FD field encountered), write down the format sequence scanned (line 5 of the algorithm), and then select the same field. The format sequence corresponding to the undetermined field of the value is compared to see if it corresponds to the same format sequence. If so, it can be inferred that the field is an FD field (lines 6-10 of the algorithm). The same format sequence here has two meanings:

1. The format sequence is the same, and there is no FD field in the format sequence;
2. The format sequence is the same, but if there are FD fields in the format sequence, it is necessary to calculate whether the values of the FD field intersect among different samples. If so, it is determined to be the same format sequence; otherwise, it is different format sequence.

Protocol state machine inference based on state annotation

Firstly, session is divided into message sequence. Using the protocol format and semantic information acquired by the above method, sample message features are extracted and clustered to obtain message type (MT) set M. Then session sample S can be expressed as message type sequence (MTS):

\[ s_i = (a_i, ..., a_n), a_i, ..., a_n \in M \]

Construct a prefix tree that accepts all \( s_i \in S \) correctly, in which the node represents the state and the edge is the input \( a_i \in M \) that causes the state transition; and then sum up the precursor type sequence P of each type m, which is expressed by regular expression.and then sum up the precursor type sequence Pi of each type \( m_i \in M \), which is expressed by regular expression:

\[ P_i = .*(r(a_j)...)r(a_j,...,a_j \in M) \]

Initial state \( q_0 \) is marked as a type without precursor type, and non-initial state \( q_i \) (i > 0) is defined as:
\[ q_i = \{m_i \mid P_i \rightarrow a_{i0}, i > 0 \} \]

Above, \( a_{i0} \) denotes the input sequence that causes the migration from \( q_0 \) to \( q_i \), i.e., state \( q_i \) is labeled as the set of all types of its \( P_i \) matchable \( a_{i0} \). Finally, Exbar algorithm is applied to fuse the states with the same label, and the minimum deterministic finite state machine is extracted.

The final minimum deterministic finite state machine contains protocol state information and context relations. Each state in the state machine represents a protocol state, while the transition relationship between states represents context relations.

4. Test result
Based on the above scheme, we try to implement the encoding. The following is the deep message parsing results under unknown protocol specifications using common HTTP messages as an example. It can be seen that useful protocol fields can indeed be extracted.

4. Concluding remarks
On the basis of researching the major protocol reverse engineering methods at home and abroad, this paper proposes an automated private protocol deep message parsing method under unknown protocol specifications, extracts protocol fields and state machines, and combines the training, analysis and optimization of network-related protocols, which can further improve the accuracy of the analysis results. The accuracy and efficiency of this method, if applied to the security monitoring of private protocol network traffic, will have a significant impact on ensuring the safe and stable operation of the power grid.

References
[1] Fan Chen, Chen Xiaochuan. Comparison of IEC 61850 and IEC 60870-5-103 Standards. Northeast Electric Power Technology, 2006(9): 47~49
[2] Tan Wenshu. Introduction of Substation Communication Network and System Protocol IEC61850. Power grid technology, 2001(25): 8～11, 15

[3] Zheng Jie, Zhu Qiang. Address Analysis and Research of Unknown Single Protocol Data Frame. computer science, 2015(11): 184～185

[4] He Yongjun, Shu Hui, Xiong Xiaobing. Reverse Analysis of Network Protocol Based on Dynamic Binary Analysis [J]. Computer Engineering, Issue 09, 2010

[5] Huang Zhou, Peng Xin, Zhao Wenyun. Reverse Recovery of Object Behavior Protocol Based on Dependency Analysis [J]. computer science, Issue 08, 2008

[6] Pan Fan, Wu Lifan, Du Youxiang, Hong Zheng. Advances in Protocol Reverse Engineering. Computer Applied Research, Issue 08, 2011