2015 International Conference on Control, Automation and Information Sciences (ICCAIS)
October 29-31, 2015, Changshu, China

Designing Snort Rules to Detect Abnormal DNP3 Network Data

Hao Li, Guangjie Liu, Weiwei Jiang and Yuewei Dai,
School of Automation
Nanjing University of Science and Technology
Nanjing , China
lihao051310@126.com ,gjieliu@njust.edu.cn,jiangweiwei_njust@163.com, dywjust@163.com

***Abstract*—Vulnerability of industrial control network communication protocol is the most important reason leading to industrial control network attacks. In this paper, the vulnerability of DNP3, the typical industrial control network communication protocol, is analyzed. The abnormal behaviors of DNP3 are categorized according to the Snort detection mechanisms. The Snort detection rule template for anomaly DNP3 data is constructed and the rules are designed according the template. The rule designing method can be generally extended to other network-based industrial control protocols.**

Keywords—industrial control network security; DNP3; protocol vulnerability analysis; snort rule template

I. INTRODUCTION

Industrial Control System (ICS) is widely used in the production process and power facilities etc. Traditionally, the security of ICS mainly depends on the network isolation. With the development of industrialization and informatization, the isolation of ICS network is gradually broken. The data exchanging between control network and enterprise intranet or the Internet is more and more frequent [1]. As a result, the network security risk of ICS is increasing. Meanwhile, with the rapid development of advanced persistent thread (APT), ICS is facing more complicated intrusion threats.

In recent years, a series of attacks against industrial control networks occur [2, 3]. For example, Shock Wave Worm virus rampant world wide in 2003, Bushehr Iran nuclear power station was attacked by Stuxnet virus in 2010 [4, 5]. After analyzing ICS security incidents recorded by RISI database, we can find that plenty of likely accidents have occurred in the power, transportation, energy, water and other fundamental industries. It also exhibits the importance of ICS network security issues [6-8].

DNP3 protocol is a typical protocol of industrial control network. It plays a significance role in ICS security to analyzes the vulnerability of DNP3 protocol. In this paper, the Snort rules for anomaly detection of DNP3 data is designed. They can be directly applied in Snort. The design method of the rule template of DNP3 can also be extended to other ICS network protocol.

This paper is organized as follows. Section 2 introduces the framework of DNP3 protocol, analyzes its vulnerability and summarizes the typical security problems. Section 3 introduces intrusion detection rules syntax and designs a set of rule templates based on Snort for detecting abnormal data flow of DNP3 protocol. Section 4 designs abnormal data flow detection rules based on the developed rule template. Section 7 summarizes this paper and discuss further issues worthy of study.

II. DNP3 PROTOCOL AND ITS VULNERABILITY ANALYSIS

A. DNP3 Drame Structure and Data Description

DNP3.0 is the international standard protocol based on the IEC TC-57 standards and developed by the IEEE Electric Power Engineering Association (PES). As shown in Figure 1, DNP3 identifies and defines three layers (physical layer, data link layer and application layer) which is called Enhanced Performance Architecture (EPA) [9]. The physical layer defines an ordinary RS-232 or RS-485 interface and fibre transceiver. The data link layer defines the information format. The application layer defines the function of DNP3 protocol.

In order to supporting the advanced function of RTU (Remote Terminal Unit) and the message with its size being greater than the maximum frame length, DNP3 defines a pseudo transport layer. The application-layer message with long size will be split into several short frames to be transmitted. Conversely, after receiving data, the short frames are assembled to complete the entire application layer message. The disadvantages caused by the small capacity of application data are resolved by the pseudo transport layer.

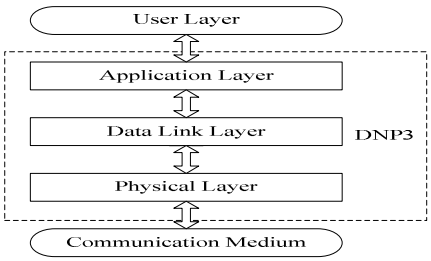

Fig. 1. The model of DNP3.0 protocol structure

1) DNP3 data link layer uses series of IEC870-5 of variable frame format: FT3. As shown in Fig 2. An FT3 frame is defined as a header block with fixed length, followed by

optional data blocks. Each block has a 16- bit CRC appended to it. The fixed header fields consist of 2-octets start, 1-octet length, 1-octet control, a destination address, a source address and a 16- bit CRC.

Fig. 2. Frame format for link layer data units

In pseudo transport layer encapsulation before user data frame, a transport layer header (TH) is added, which contains the information for reconstructing the entire message. All pseudo-transport layer messages have a TH as shown in Fig3.

Fig. 3. Pseudo transport layer data unit format

TH : Transport control octet. One octet in length.

Application Layer Data Block: 1 to 249 octets in length.

When an application requests the transmission of a long message, the message is divided into small slices to fit in a single DNP 3 data link frame.

2) DNP3 protocol defines the format of the application layer messages (APDU). The master station is defined as the station which sends a request message while the outstation is the slave device. RTU or intelligent terminals (IEDS) sends a response message which is specified in advance. In DNP3, master stations just send application layer request messages and outstations just send application layer response messages. The layout of the application layer message is shown in Fig5.

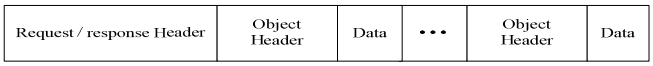

Fig. 5. Application layer packet format

Request Header identifies the purpose of the message and consists of application protocol control information and function code. Response Header contains the same information as Request Header pluses an additional field containing internal indications of the outstation. Object Header identifies the data objects that follow and consist of Object field, Qualifier field and Range field.

DNP3 uses Object Group and Variants to describe different data and events in practical application. Each Object Group defines one type of data, Variant describes a specific form of the data type. Both of them have a specified format and number. These datum can be classified into 4 categories: Class 0-3 with corresponding priority assigned. The application layer uses the object-oriented data structure to guarantee that it can adopt the following operation methods flexibly: (1) Active reporting mode; (2) Polling mode; (3) Variable Polling mode and (4) Peer-to-Peer transmission mode.

B. DNP3 Protocol Vulnerability Analysis

DNP3 protocol is mainly used in the industrial control network isolated from other computer networks. Except the requirements for real-time, reliability, and efficiency, the security of the protocol is not considered strictly, and the protocol management organization has not added any security features to DNP3 protocol so far. The current typical security problems in DNP3 protocol are the following [10].

1) Lack of Authentication Protection

In the course of the communication of DNP3 protocol, there is no relevant definition of any authentication. An attacker can easily disrupt the entire or partial control process by creating a conversation using well-defined function code and data type.

2) Lack of Authorization Protection

DNP3 protocol has no access control mechanism. As a result, any user can perform any operation to run an arbitrary function.

3) Lack of Encryption Protection

Encryption ensures the confidence of information. During the process of DNP3 protocol communication, the address and the command are all transmitted in plain text, so datum can be easily captured and parsed by attackers. It might be helpful to analyze the industrial control object and process.

Apart from the security problem caused by the design flaws, DNP3 protocol has the other serious security problems. For instance, illegal user's malicious tampering and abuse of user protocol function code. Once obtaining the authority to access the network, the attacker can send any function code to the server or send false information to the client to get useful feedback.

In recent years, the continuous emergence of industrial network security incidents indicated that the abuse of DNP3 function code is the main reason of attacks. Considering the compatibility of the security mechanism and the existing industrial control systems, a security protection mechanism for DNP3 application layer data is presented, which monitors the process of DNP3 communication, analyzes the application layer data flow, and makes real-time detection to discover the abnormal data flow.

Based on the vulnerability analysis of DNP3 protocol [11], the typical abnormal behaviors of DNP3 are summarized in Table I-V. The abnormal behaviors are divided into five categories: Non-standard protocol, Potential scanned, Potential Denial of Service, Denial of Service attack and Mixed attack. Meanwhile, the detection parameters and corresponding descriptions of each kind of the abnormal behavior are given in Table I-V.

TABLE I. NON_STANDARD PROTOCOL

Serial Number	Parameters	Abnormal Behavior Description
1	The start of data link layer header	Run the non DNP3 protocol on the DNP3 port
2	Data link layer function code	Illegal request / response function code, the range beyond (0x0F,0x00)

3	Serial number of application layer message header	The illegal message segment number, the range beyond (0x0F,0x00)
4	Application layer request function code	Illegal request function code, the range beyond(0x00,0x21)
5	Application layer response function code	Illegal response function code, the range beyond{0x81,0x82,0x83}

TABLE II. POTENTIAL SCANNED

Serial Number	Parameters	Abnormal Behavior Description
1	Abnormal function code 81	Configuration scan(defined point column and its value10/min)
2	Abnormal function code 82	Available function code scanning(3 /min)

TABLE III. POTENTIAL DENIAL OF SERVICE

Serial Number	Parameters	Abnormal Behavior Description
1	Active response function code 0x82	Stop active response
2	Active response function code 0x82	Repeat a large number of active response packages to the DNP3 server
3	Function code 13(0x0D)	Client repeat cold start
4	Function code 14(0x0E)	Forced DNP3 server hot restart

TABLE IV. DENIAL OF SERVICE ATTACK

Serial Number	Parameters	Abnormal Behavior Description
1	Function code 13(0x0D)	The unauthorized client to repeat the cold start
2	Function code 18(0x12)	Stop the application on DNP3 server

TABLE V. MIXED ATTACK

Serial Number	Parameters	Abnormal Behavior Description
1	Read information function code 01	Unauthorized client to read information from PLC and other devices
2	Write information function code 02	Unauthorized client to rewrite the information from PLC and other devices
3	Abnormal function code 05,64	The unauthorized client to other requests for other devices such as PLC
4	Valid server address	Broadcast request for client
5	Valid server address	Broadcast request for unauthorized client

III. SNORT RULE TEMPLATE FOR DNP3 ANOMALY DETECTION

A. Snort Intrusion Detection Rule Syntax

Snort is a lightweight and open source network intrusion detection system. It has three operation modes [14].

1) Packet sniffer mode: The network card is set to the mixed mode to capture packets and analyze the network data.

2) Packet record mode: All packet information are recorded in a text format to the directory specified by the user.

3) Network intrusion detection mode: After the configuration file named Snort.conf is loaded, packets are analyzed and checked whether matching with the intrusion detection rules. If the packets match with the rules, the alert mode will be launched to output warning information.

The structure of Snort rule is shown in Fig. 6. It is composed of two parts: the rule header and the rule body [13].

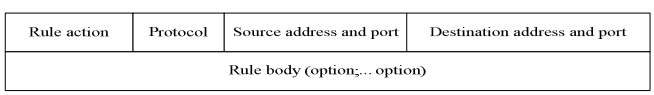

Fig. 6. Snort rule structure

The rule header contains actions initiated after successful matching with the rules, such as Alert, Log, Pass, Activate, Dynamic, etc. The packet matching conditions are also included in the rule header, such as protocol type, source address, source port, destination address and destination port. The rule body is composed of option keywords and contents. Keywords and contents are separated by a colon, and options are separated by a semicolon. The rule option contains intrusion feature string (content), alert content (msg), load length (dsize), type (classtype), priority (priority), version (rev), and important information relevant to the mode matching function (nocase, offset, depth, etc.). Snort rule library is similar to a virus feature library. Each rule is related to one or one kind of network attacks. So users can define new rules and add them to the rule sets, such as ddos.rules for DDOS attack. In addition, users can also make necessary adjustments of existing rule base files, and then add the new rule file to the Snort.conf configuration file.

B. Rule Template Design of Snort

The traditional preparation of Snort rules is based on an effective flow feature, so in the actual preparation process for rules, the unique characteristics of the flow must be found, and rules based on these characteristics should be written manually. According to the vulnerability analysis of the DNP3 protocol, this paper also summarizes and classifies the typical feature set (abnormal behaviors). The rule header template and rule body template is designed respectively, so

that the preparation of rules can be easily finished by loading feature set into configuration file.

1) Rule Header Template Design

According to the above discussion, the rule head is divided into seven parameters in the design of rule head template, and each parameter value corresponds to a string array as shown in Table VI.

Firstly, five pre-set rule actions should be initialized in Action array, which are named Pass, Log, Alert, Activate and Dynamic. Discard, record, warning, activation of other linkage rules and other actions should be adopted respectively to deal with abnormal network flow. Snort supports total 4 protocol types which should be initialized in the Protocol array including IP, ICMP, TCP and UDP. If it is an IP protocol, the source/destination address and port parameters of the rule header are respectively set in the Snort.config configuration file, and the contents of each parameter in the array can also be added or deleted according to the actual ICS environment. Flow direction array named Direction which includes three basic operators between source and destination. Its function is monitoring the server and the client's data flow, respectively. A specific initialization for each array of parameters is defined as follows.

```
String Action[5]={“pass”,”log”,”alert”,”activate”,
”dynamic”};
string Protocol[4] = {“ip”,”icmp”,”tcp”,”udp”};
string Src_address[2] ={“any”,”$DNP3_CLIENT”,
”$DNP3_SERVER”};
string Src_port[2] ={“any”,” $DNP3_PORTS”,”
$DNP3_SERVER”};
string Direction[3] = {“->”.”<-”,”<>”};
string Dst_address[2] ={“any”, ”$DNP3_CLIENT”,”
$DNP3_SERVER”};
string Dst_port[2] = {“any”,” $DNP3_PORTS”};
int i = 0,j = 0,sa = 0,sp = 0,dr = 0,da = 0,dp = 0;
bool Rq = false,Rs = false;
```

2) Rule Body Template Design

The Snort rule body is in a pair of parentheses after the rule header, which belongs to the optional content. The function provides further analysis for the detection engine based on the rule header information. As shown in Table VII, the rule body is divided into ten effective parameters in the rule body template design.

In order to distinguish the packets which are sent from the client or the server, the Flow array is initialized with 8 options. According to classification of typical abnormal behaviors of DNP3, rule class Classtype array contains five rule types, and each abnormal behavior has a default priority. The remaining parameters can be determined according to different abnormal behavior descriptions, for the time being, the initialization is an empty value or 0. Rule ID and rule version number should be determined by the administrators of intrusion detection system when they write the rule system. A specific initialization of each parameter is defined as follows.

```
String Flow[8]={ “to_client”, “to_server”, “from_client”,
“from_server”, “established”,“stateless”,“no_stream”,
“only_stream”};
String Classtype[4]={“non_standard_protocol”,
“attempted_recon”,“attempted_dos”, “successful_dos”,
“misc_attack”};
String  Content = null, Msg = null,Sid = null;
int  Offset = 0,Depth = 0,Rev = 0,Priority = 0;
```

IV. DNP3 ANOMALY RULES DESIGN FROM TEMPLATES

A. Snort Detection Rule Classification for Anomaly Behavior

According to the classification and description of the typical abnormal behavior of DNP3, the design of the following Table VIII-XII correspond to Table I-V. Snort rule syntax is used to confirm corresponding parameters for each kind of anomaly behavior. The generation phase of each anomaly behavior is analyzed and the request/response identifier is set, and then the priority for each detection rule is determined according to the severity of the different abnormal behavior.

TABLE VI. RULE HEADER TEMPLATE

Action	Protocol	Source address	Source port	Direction	Destination address	Destination port
Action[i]	Protocol[i]	Src_address[sa]	Src_port[sp]	Direction[dr]	Dst_address[da]	Dst_port[dp]

TABLE VII. RULE HEADER TEMPLATE

Communication flow direction	Matching feature string	Matching starting position	Matching maximum depth	Matching threshold	Message	Classtype	Rule ID	Rule version	Priority
Flow[i]	Content	Offset	Depth	Threshold	Msg	Classtype[j]	Sid	Rev	Priority

TABLE VIII. NON_STANDARD PROTOCOL

Serial Number	Abnormal Parameter Values	Request / response	Priority
1	pcre: “/(?!\x05\x64)/iAR”	Rq=true Rs=true	2
2	(byte_test:1,>,15,4)	Rq=true Rs=true	1
3	(byte_test:1,>,15,12)	Rq=true Rs=true	1
4	(byte_test:1,>,33,12)	Rq=true Rs=false	1

5	(byte_test:1,<,129,12)&(byte_test:1,> ,131,12)	Rq=false Rs=true	1

TABLE IX. POTENTIAL SCANNED

Serial Number	Abnormal Parameter Values	Request / response	Priority
1	content:"|81|"; offset:12;depth:1; pcre:"/[\S\s]{1}(\x02|\x04|\x06|\x0a|\x0c|\x0e)/iAR";	Rq=true Rs=false	2
2	content: "|81|"; offset:12; depth:1; pcre:" /[\S\s]{1}(\x01)/iAR";	Rq=true Rs= false	2

TABLE X. ATTEMPTED_DOS

Serial Number	Abnormal Parameter Values	Request / response	Priority
1	(content:"|15|"; offset:12; depth:1;)	Rq= false Rs=true	2
2	(content:"|82|"; offset:12; depth:1; threshold: type threshold, track by_src , count 5, seconds 10;)	Rq=true Rs= false	2
3	(content:"|0D|"; offset:12; depth:1;)	Rq= false Rs=true	2
4	(content:"|0E|"; offset:12; depth:1;)	Rq= false Rs= true	2

TABLE XI. SUCCESSFUL_DOS

Serial Number	Abnormal Parameter Values	Request / response	Priority
1	(content:"|0D|"; offset:12; depth:1;)	Rq= false Rs=true	1
2	(content:"|12|"; offset:12; depth:1;)	Rq= false Rs=true	2

TABLE XII. MISC_ATTACK

Serial Number	Abnormal Parameter Values	Request / response	Priority
1	(content:"|01|"; offset:12; depth:1;)	Rq= false Rs=true	2
2	(content:"|0564|";depth:2; pcre:"/[\S\s]{10}(\x02|\x04|\x05|\x06|\x09|\x0A| x0F|\x12)/iAR")	Rq= false Rs=true	1
3	(content:"|0564|";depth:2; pcre:"/[\S\s]{10}(\x03|\x07|\x08|\x0B|\x0C|\x10|\x11|\x13|\x14|\x15|\x16|\x17|\x18|\x19|\x1A|\x1B|\x1C|\x1D|\x1E)/iAR";)	Rq= false Rs=true	1
4	(content:"|FF FF|"; offset:4; depth:2;)	Rq= false Rs= true	2
5	(content:"|FF FF|"; offset:4; depth:2;)	Rq=false Rs=true	1

B. Rule Header Design for Anomaly Detection

According to the above classification of detection rules, rule header action type should be determined by the value of priority in the typical anomaly behavior table, and different priority corresponds to different rule action. According to the state of request/response (Rq/Rs) in the table, subscript value of other parameters array in the rule header should be judged, and the exact value of each parameters in the rule header should also be determined. The writing of head rule statement can be finally completed. The subscript judgment pseudo code of each parameter array is as follows.

```
if(priority == 1                    if(Rq && !Rs)
||priority == 2)                    sa=2;sp=1;dr=0;da=1;dp=0;
i=2;                                if(!Rq && Rs)
if(priority == 3)                   sa=1;sp=0;dr=0;da=2;dp=1;
i=1;                                if(Rq && Rs)
else                                sa=0;sp=0;dr=0;da=0;dp=1;
i=0;
```

The following example is the preparation of rule header about Non_standard protocol in Table VIII (Serial number 1) . The protocol type is TCP, Protocol[2], priority=2. Rule action in rule header template is determined as Action[2]=alert. The source address is determined as Src_address[0], source port as Src_port[0], flow direction as Direction[2], destination port as Dst_address[2], destination port as Dst_port[1] when the request / response identifier (Rq/Rs) is true. The rule header is designed as follows.

```
alert tcp any any <>$DNP3_SERVER $DNP3_PORTS.
```

C. Rule Body Design for Anomaly Detection

According to the classification of the detection rules, the contents of the keyword classtype in the rule body should be identified according to the classification of each abnormal behavior. Then the option content of the keywords flow, content, offset, depth, msg, and priority should be determined according to values of each classtype in the table. When the keyword msg is output, it corresponds to anomaly behavior description. The keyword priority corresponds to priority in the table. Finally, a unique ID and version number should be assigned for each rule. The contents for the keyword sid and rev also should be determined. Finally, the direction for communication flow should be determined according to request/response signs, and the pseudo-code to determine the content of the keyword flow in communication flow is as follows.

```
if(Rq && !Rs)
   flow: Flow[3], Flow[4];
if(!Rq && Rs)
   flow: Flow[2], Flow[4];
if(Rq && Rs)
   flow: Flow[4];    msg:Message;
```

Anomaly feature string matching is the most important and complex link in Snort rule body. The writing form and parameters of the keyword of content determine the integrity and effectiveness of the detection rules. By extracting the value of abnormal parameter value in table, the following example is given to explain the writing rules for the feature string matching and pattern matching.

```
Turn off the active response (Table XI- serial number
1):content:"|15|"; offset:12; depth:1;
Set the detection threshold within 60 seconds to matching 30
times(Table XI- serial number 2):threshold:type
threshold,track by_src, count 30,seconds 60.
```

D. Anomaly Data Detection Rule Design

According to the designed intrusion detection rule template, the design of detection rule for a typical potential denial of service attack (repeat a lot of active response packets to the DNP3 server in a short time) is given as the practical example. Through the function code analysis of DNP3 protocol application layer, the function code 0x82 is generally used to provide feedback of the alert information or equipment update information to the server. An attacker can use packet builder or DNP3 simulator to generate a number of active response packets, and send a large number of active response packets repeatedly to the DNP3 server in a short time, which can not only cause delay of normal response packet, but also increase the load of the entire industrial system, thus it is a potential deny service attack.

Because the detected protocol type is TCP protocol, in the rule header, Protocol[2]=tcp, priority=2, Action[2]=alert; The source address/destination address, source port Src_port[0], flow direction Direction[0], destination port Dst_port[1] should be set according to the request/response identifier (Rq/Rs).

According to Table X, the contents of the keyword classtype in the rule body are determined. Then the priority option content of keywords flow, content, offset, depth, msg and priority should be determined according to all parameters with the serial number 2. The output information of the keyword msg is determined by the list of abnormal behavior description (Table III serial number 2). Finally, a unique ID and version number for each rule is set. The contents of the keyword Sid and rev should be determined. Finally, the rule is written as follows.

```
Alert tcp $DNP3_SERVER $DNP3_PORTS
->$DNP3_CLIENT any
(flow:established; content:"|82|"; offset:12; depth:1;
msg:"Unsolicited Response Storm!";
threshold: type threshold, track by_src, count 5, seconds 10;
classtype:attempted-dos; sid:1111203; rev:1; priority:2;)
```

V. CONCLUSION

Protocol security of industrial control system was analyzed and studied in this paper, but in general, the research oriented industrial intrusion detection system is still in the initial stage, especially in China. In order to explore the anomaly detection in ICS, the implementation principle of DNP3 is introduced and the corresponding security threat analyzed. A rule template based of Snort anomaly data flow detection is designed for DNP3. The detection rule can be designed according to the template and the abnormal behaviors. In future work, the design method proposed in this paper will be extended to Modbus, PROFIBUS and other industrial control protocols.

REFERENCES

[1] Finnan, Kevin L. "SCADA Networking Facilitated Using DNP3" J. Pipeline & Gas Journal,2011,2382.

[2] Gao G. "Siemens had industrial system vulnerabilities or influence most industrialized countries" N. Southern Daily, 2011-6-8 (A18) (In Chinese).

[3] Bencsáth B, Pék G, Buttyán L, et al. Duqu: "Analysis,detection,and lessons learned" C. //ACM European Workshop on System Security (EuroSec). 2012, 2012.

[4] Peter Beaumont."Stuxnet worm heralds new era of global cyberwar" N. London: Guardian.co.uk,2010-9-30(16).

[5] Kris Ardisk. "Stuxnet caused the embedded system security considerations" J. Journal of Electronic Design Technology, 2013 (3) : 49 and 50. (In Chinese).

[6] Bradley Reaves,Thomas Morris. "An open virtual testbed for industrial control system security research" J. International Journal of Information Security,2012,114.

[7] H.M. Leith,John W. Piper. "Identification and application of security measures for petrochemical industrial control systems" J. Journal of Loss Prevention in the Process Industries,2013,266.

[8] Barry C. Ezell,R. Michael Robinson,Peter Foytik,Craig Jordan,David Flanagan. "Cyber risk to transportation, industrial control systems, and traffic signal controllers" J. Environment Systems and Decisions,2013,334.

[9] DNP Users Group DNP PRODUCT DOCUMENTATION .

[10] LI H. "Industrial control system and its security research report" R. Green Science and Technology, 2013.

[11] LU H. "Industrial control system vulnerability testing and risk assessment research" D. Shanghai: East China University of Science and Technology, 2014. (In Chinese).

[12] Sanghyun Park,Kyungho Lee,Sang-Soo Yeo. "Advanced Approach to Information Security Management System Model for Industrial Control System" J. The Scientific World Journal,2014.

[13] Roesch Martin, Green Chris. Snort users manual2.9.6 EB/OL. 2014-09-25.

[14] Zhang M. "Design and implementation on a distributed intrusion detection system based on Snort" D. Snort Beijing University of Technology, 2012 (In Chinese).

[15] Farhad Nabhani,Todd Mander,Simon Hodgson,Paul Shelton. "Critical infrastructure protection security layer for DNP3 devices".J. IJMR,2012,7.