Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 73-90
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Research on DoS Attack Detection Method of Modbus TCP in OpenPLC

Tongxin Li[1], Yong Wang[1], Cunming Zou[2], Yingjie Tian[3], Lin Zhou[1], Yiwen Zhu[4]

[1]College of Science, Shanghai University of Electric Power, Shanghai, China
[2]National Quality Supervision and Testing Center of Security Products for Network and Information Systems, The Third Research Institute of Ministry of Public Security, Shanghai, China
[3]Power Big Data Center, State Grid Shanghai Municipal Electric Power Company Electric Power Research Institute, Shanghai, China
[4]Shanghai Cloud Sword Information Technology Co., Ltd., Shanghai, China

Email: 18351801151@163.com

How to cite this paper: Li, T.X., Wang, Y., Zou, C.M., Tian, Y.J., Zhou, L. and Zhu, Y.W. (2021) Research on DoS Attack Detection Method of Modbus TCP in OpenPLC. *Journal of Computer and Communications*, **9**, 73-90.
https://doi.org/10.4236/jcc.2021.97007

Received: January 25, 2021
Accepted: July 27, 2021
Published: July 30, 2021

Abstract

With the development of new information technologies such as cloud computing, Internet of Things, and mobile Internet of Things, Industry 4.0, Smart Manufacturing and Made in China 2025 have been proposed as the main content of the development of the next industrial revolution. In order to realize these projects with the common characteristics of intelligence, service, and green, a new manufacturing model, digital twin, is proposed, which combines the digital twin with industrial systems, that is, the industrial control virtualization system. However, due to the frequent occurrence of industrial control system security incidents in recent years, the industrial control virtualization system is vulnerable to attacks. The industrial control system is huge and cumbersome. Once attacked, it will cause consequences that affect the whole body. In response to this problem, this article carried out a research on DoS attack detection methods for Modbus TCP in OpenPLC, using OpenPLC as a tool for industrial control system virtualization, building a digital twin system with Raspberry Pi, and launching DoS attacks on the system, combined with Snort Intrusion detection is carried out, and the experimental results show that the built digital twin system can detect DoS attacks in OpenPLC.

Keywords

Digital Virtualization, Communication Protocol Vulnerability, OpenPLC, Snort

1. Introduction

The concept of industrial control system virtualization is mainly derived from the term Digital Twins. With the development of new information technologies such as cloud computing, Internet of Things, and mobile Internet of Things, Industry 4.0, smart manufacturing and Made in China 2025 have been proposed as the main content of the next industrial revolution. In order to realize these projects with the common characteristics of intelligence, service and green, a new manufacturing model-digital twin is proposed. And digital virtualization is an important part of it. Virtualizing the industrial control system can turn the complex into "simplicity".

With the rapid development of technology, the industry needs to adjust quickly, and improve product efficiency, increase output and reduce costs. Due to the high cost of purchasing machines and procedures when developing Industry 2.0 and Industry 3.0, most small and medium-sized enterprises are developing toward Industry 4.0. Development planning must be carried out before actual development. Different forms of production line planning are also different. The digital twin system can be applied to this, not only can display real data on the production line in real time, but also display the status of the production line, and predict the situation that will occur, helping small and medium-sized companies to transform existing production into full automation. In addition, digital twins can be used to create and plan tools compatible with the system, so as to easily directly connect the production line system with customer needs, and gradually develop an automated system for future use.

Gartner, one of the most authoritative IT and consulting companies in the world, has listed digital twins as one of the top ten strategic technology development trends from 2017 to 2020 [1] [2] [3] [4]. They believe that with the integration of technology and people, there will be many opportunities to create digital versions of physical systems, which will represent different objects in the real world. Digital twins are mainly used in the field of industrial manufacturing. More and more well-known domestic and foreign companies have begun to study the application fields of digital twin technology, such as product design, manufacturing and services [5]. The digital virtualization system also has the characteristics of a digital twin. The key of the digital twin is to connect the physical world and the digital virtual world together and realize the two-way dynamic interaction of information flow. On the contrary, there is a term called Digital Shadow, which means that there is only one-way data flow between the state of the physical object and the digital object. The digital twin can be used to perform tasks, but the control information must be fed back into the system. Virtualizing the industrial control system can predict the failures that will occur in the system, prepare emergency plans in advance, optimize the system, and increase the service life.

The industrial control system is digitally virtualized to completely and truly produce the entire industrial control system. However, with the frequent occurrence of industrial control system security incidents in recent years, industrial control virtualization systems are vulnerable to attacks. The industrial control system is huge and cumbersome. Once attacked, it will cause consequences that affect the whole body. In order to improve the security of industrial control virtualization system, there are mainly the following problems [6]:

1) There are many types of industrial control virtualization systems, but they lack security measures;

2) Some physical environments of the industrial control system are set up in places where they cannot be operated, such as in the wild, and cannot be maintained immediately after being attacked, and the maintenance is difficult.

Aiming at the above problems, this article uses the cloud server as the communication host, the Raspberry Pi as the communication slave, uses the Modbus communication protocol for communication, and is equipped with OpenPLC to simulate a small industrial control system, build a digital virtualization system, and conduct attack tests on the system. Use the Pfsense firewall to add snort as a plug-in for detection, thereby improving the security of the industrial control system.

2. Related Research

The concept of digital twins was first proposed by Micheal Grieves of the University of Michigan in 2003, but it was not clearly valued. With the cooperation between the US Air Force Research Laboratory and NASA in 2011, the concept of digital twins for aircraft was proposed, brings a clear definition to the digital twin. With the continuous introduction of technology and development strategies such as Industry 4.0 and smart manufacturing, digital twins have begun to receive attention from all walks of life. Many products and research based on digital twins have also emerged.

In order to promote the improvement of laboratory equipment management methods, the Chinese Academy of Sciences established the SAMP platform, but there was a problem that facility failures could not be actively predicted. Ma Yue *et al.* [7] used digital twin technology to build a virtual laboratory platform and its digital twin experiment the laboratory model mainly includes four or five points of physical laboratory, virtual laboratory, laboratory twin data, and digital twin laboratory service platform. It can predict, maintain and improve the system while ensuring the normal operation of the laboratory physical platform, and promote the experiment Digital construction of the laboratory platform. In order to adapt to the new development trend, the Beihang digital twin technology research team proposed a digital twin five-dimensional model [8]. In the original three-dimensional model, that is, physical entities, virtual entities and the connection between the two, services and twin data are added. Tao Fei and others also designed a digital twin workshop [9], whose main features are virtual and real integration, data-driven, full elements, full processes, full business integration and integration, iterative operation and optimization, etc. The key technology lies in the physical workshop “human-Machine-things-environment” interconnection and inclusive technology, virtual workshop construction, operation and verification technology, and workshop twin data construction and management technology. Based on the feature description of the workpiece digital model, Chen Moran *et al.* [10] established a digital twin model, optimized the nearest neighbor feature matching algorithm, combined with the Hough voting mechanism, and improved the accuracy of finding the location of the landmark in the twin model. Based on digital twin technology, Liu Zhifeng *et al.* [11] proposed a new method to solve the scheduling problem of parts intelligent manufacturing workshop, and called it a scheduling cloud platform, which uses real-time monitoring data and big data analysis to predict workshop production And diagnosis, and put the proposed model on the ground for verification. Wang Shilong *et al.* [12] proposed a new paradigm based on hierarchical digital twin industrial Internet manufacturing-fog manufacturing, and gave the definition of fog manufacturing, and analyzed the difference between fog manufacturing and traditional industrial Internet and cloud manufacturing. In the above, a digital twin-based man-machine-object fusion drive control system is proposed, and a typical machining application is used as a case to verify the feasibility and effectiveness of the proposed fog manufacturing model.

Attack detection is a gradually developing research. Fuzzing is a technique that attempts to discover security vulnerabilities by sending random input to an application or device. Therefore, it is widely used to test input validation and security vulnerabilities in application logic. In 2015, Qi Xiong [13] and others proposed an intelligent fuzzing technology for Modbus-TCP, described in detail the architecture of the technology, and proposed an adaptive test case generation algorithm and test process workflow. Test cases can be generated intelligently based on target feedback.

Building a real-time test bed is also one of the research methods. In 2015, Bo Chen *et al.* [14] proposed a real-time physics framework test bed that integrates real-time power system simulators and communication system simulators to study the vulnerability of networks and physical systems in smart grids. The paper uses Opnet's semi-physical simulation system (SITL) and open source Linux tools and server implementation, and discusses the results of two network attacks on Modbus/TCP protocol, and improves the protocol attack detection.

3. Problem Analysis

3.1. OpenPLC Communication Process

OpenPLC is an open source PLC control platform [6], which is created based on the actual programmable logic controller architecture on the market. It is a modular system with extended functions. OpenPLC can currently run on a variety of devices such as Raspberry Pi. **Figure 1** shows the communication topology of OpenPLC, and you can also see its system structure and the modules it contains: network server, MatIEC compiler, network layer, hardware layer, etc.

OpenPLC is a real-time program, which allows uploading and compiling PLC programs. It always runs on port 8080 and can be opened in most web browsers. The network layer is responsible for communicating with SCADA. Currently,

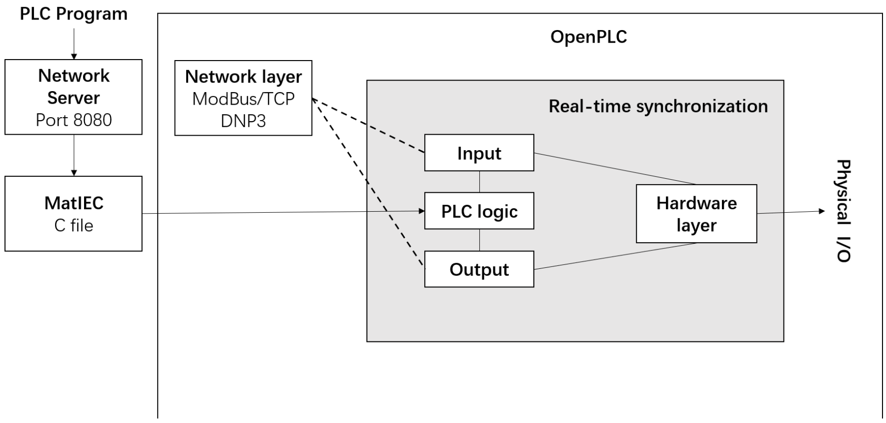


Figure 1. OpenPLC communication topology.

OpenPLC supports Modbus/TCP and DNP3. By default, Modbus is used on port 502 and DNP3 is used on port 20,000.

After OpenPLC is installed on the Raspberry Pi, it will automatically run as the Raspberry Pi starts. Open the Raspberry Pi operation page on the web and you can add the corresponding communication protocol and connected slave stations. After starting OpenPLC, you can see the operation of the protocol in its log.

3.2. Digital Twin System

Digital virtualization, that is, digital twins, has been widely used, but there are few mentions about the network security of digital twins. Once an attacker attacks the virtualized system, the virtual system's monitoring of the physical system will fall short, and the physical system will be in a situation of being "exiled" and unable to obtain its state synchronously. This is for the use of digital twins. In terms of factories and companies, it is very dangerous.

The communication protocols generally used in industrial control systems include Modbus/TCP, Ethernet/IP, DNP3, etc. These protocols were originally used in a closed and trusted network, so safety issues were not considered. The industrial control virtualization system also has the same defects as follows [15]:

1) The intelligent system faces systemic risks. The digital twin system is the connection between the physical system and the virtual system. On the one hand, the basic equipment and control system of intelligent manufacturing will face unknown network risks; on the other hand, the intelligent manufacturing system may face data security risks.

2) The development of digital twin cities cannot be ignored. Critical information infrastructure risks exist. Secondly, a large number of new technologies and applications bring unknown risks, and the corresponding new services and security management mechanisms are also lagging to varying degrees.

Therefore, in combination with the current situation and the problems faced, this solution is proposed, and on the basis of the digital twin system, safety prevention and control measures are built in order to effectively solve the current problems.

3.3. Attack Detection Method

This article first built a small industrial control system, and then connected, completed the communication of the system. Secondly, build the digital twin system. After the system is built, perform corresponding port scanning, data capture, data control, and data analysis. The main process is shown in Figure 2.

To realize the digital twin, OpenPLC is installed on the server. OpenPLC can virtualize an industrial control system. By adding slaves, different terminals can be added to the monitoring range to achieve the purpose of monitoring the entire industrial control system. This article uses Raspberry Pi and PC as the terminal to communicate with the server by modbus. It is designed for learning computer programming education, and its system is based on Linux. In order to achieve the connection between the external network and the internal network, install peanut shells on the internal network host to perform internal network penetration, and assign a public network IP and port to the Raspberry Pi to connect to the server. The server configured with OpenPLC can pass RunTime software, to program, monitor, defend and deploy PLCs such as Raspberry Pi.

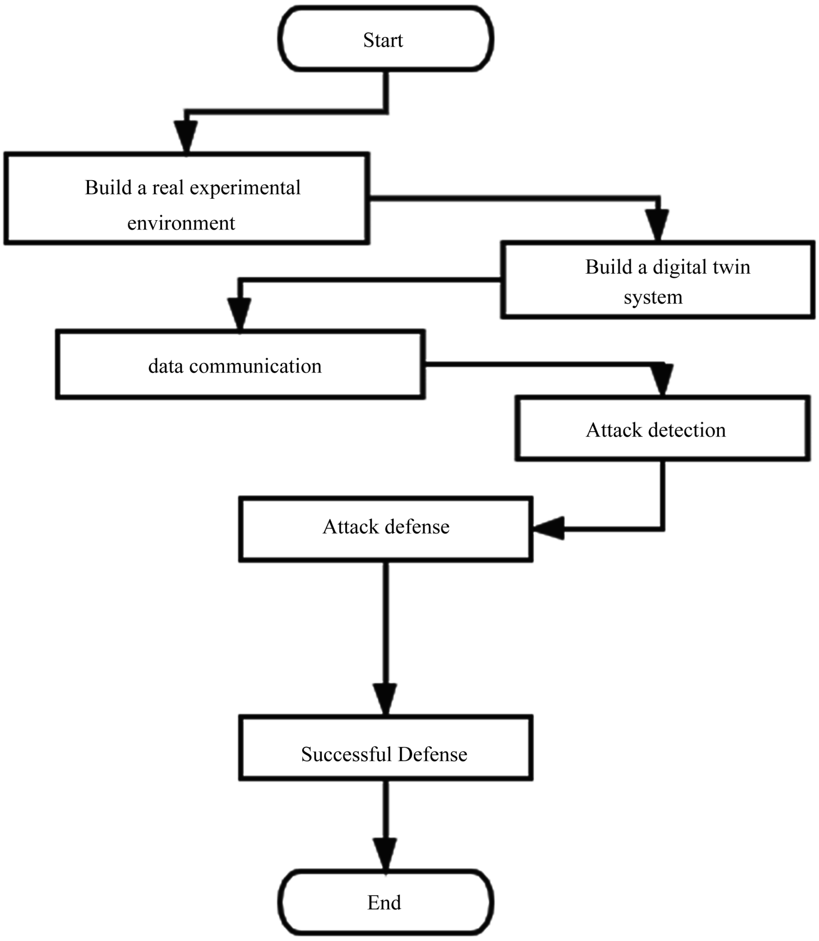


Figure 2. Attack detection process.

In the industrial environment where the digital twin system is built, the pfsense firewall configured with snort is deployed in a wide area network (WAN) environment. In the industrial control system environment, devices in the same network environment are protected by the pfsense firewall. Install the snort plug-in in pfsense, configure the interception rules, detect the WAN interface, and take defensive measures. Then, when an attacker conducts a DoS attack, it can be caught by snort and issued an early warning to prevent it. It can detect external attacks, and can block other irrelevant devices including attackers. Effectively guarantee the communication security of the industrial control system under the WAN environment. Such a network topology diagram is shown in Figure 3.

3) Accept the problem of discontinuous serial numbers: There are three message formats in the IEC 104 protocol, namely I format for effective data transmission, the S format for number confirmation, and the U format for connection maintenance. I frame is divided into two parts, APCI and ASDU, collectively referred to as APDU [13], while S frame and U message have only APCI part.

When receiving an I format data frame, compare the sending sequence number with the local receiving sequence number, if they are the same, then receive it, otherwise decide whether to discard it according to the situation. This guarantees data to a certain extent However, when the network delay is large, the application layer data packets may arrive out of order. When a data packet is discarded due to out of order, subsequent messages will be out of order. The way to solve the problem of serial number discontinuity is to adopt the window receiving method, as shown in Figure 3.

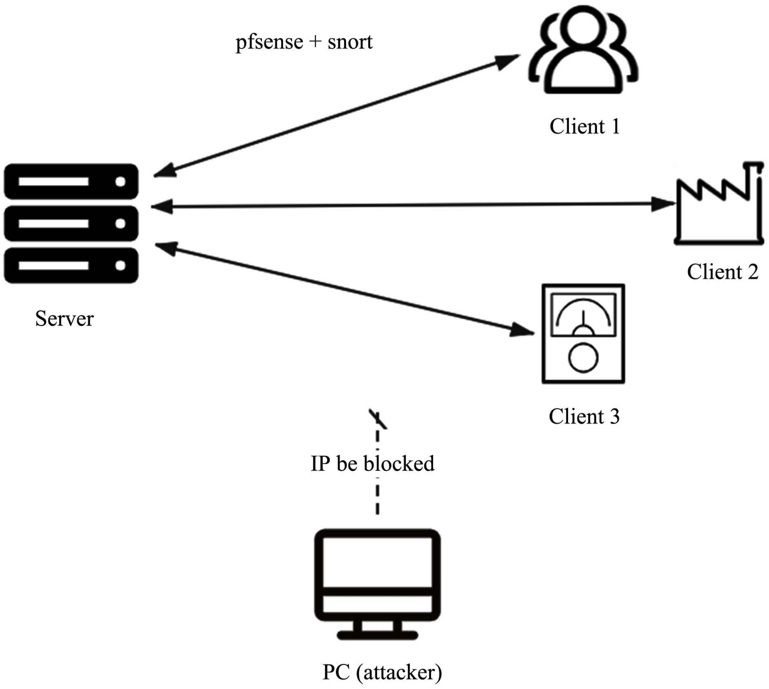


Figure 3. Schematic diagram of window receiving mode.

In order to enhance the confidentiality, integrity, availability and controllability of the communication data under the digital twin, an attack detection algorithm based on this system is proposed for typical DoS attacks, which successfully defends against DoS attacks and provides more security for the industrial control system.

The attack detection algorithm uses the blacklist function of the pfsense firewall and snort rule matching to achieve attack detection. The detailed steps are as follows:

1) Unknown visitors access/attack the system;

2) Pfsense firewall activates the black list function;

3) Compare the IP in the blacklist, if the visitor's IP is in the blacklist, then restrict its access and end this process; if the visitor's IP is not in the blacklist, then enter the next process;

4) Match with snort rules, if the set rules cannot be matched, then let go and end the process; if the set rules are matched, the IP will be blocked (blocked);

The attack detection algorithm flow is shown in Figure 4.

The Algorithm 1 is described as follows, where the unknown visitor is represented by A, and whether the data packet meets the blacklist is represented by B.

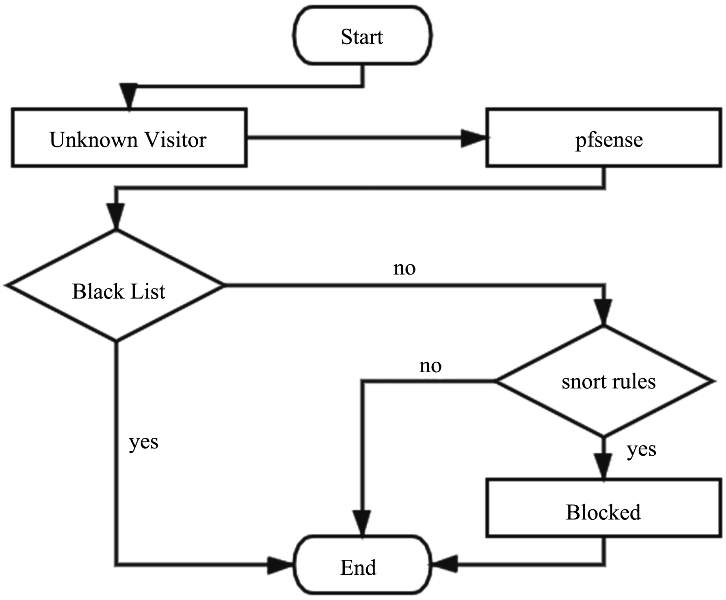


Figure 4. Attack detection algorithm flow.

Algorithm 1. Attack detection algorithm.

```
Input: Unknown Visitor A
Output: Attack detection result
1) Run pfsense;
2) Run snort and add rules;
3) If IP(A) == IP (Black List), block the visitor A;
4) If A matches the snort rules, an alert is issued and blocked ← IP(A).
The algorithm can successfully detect the DoS attack initiated by the attacker, add the attacker's IP
to the block list, and successfully block the IP.
```

4. Safety Analysis

4.1. Experimental Environment

The experiment consists of a cloud server, a Raspberry Pi 3B+ with Debian installed, a router, and a terminal computer. The physical connection of the hardware devices is shown in Figure 5.

The snort version is 2.9, the OpenPLC version is OpenPLC_v3, and the HMI version is ScadaBR. The host IP is 192.168.0.123, and the attacker's IP is 192.168.0.1. Table 1 describes the environmental configuration involved in the experiment.

After completing the physical connection, the cloud server acts as the Modbus master and the Raspberry Pi acts as the Modbus slave, communicating through the modbus TCP protocol.

Deploy OpenPLC on the server and use the OpenPLC editor to write programs to completely restore the physical system, thereby realizing the monitoring of the industrial control system. The ladder diagram settings involved in the experiment are shown in Table 2. Two variables are set, namely control and lamp. Both variables are of BOOL type and set their initial value to FALSE. The position information indicates that the pin positions on the Raspberry Pi are %IX1.5 and %IX1.6 respectively.

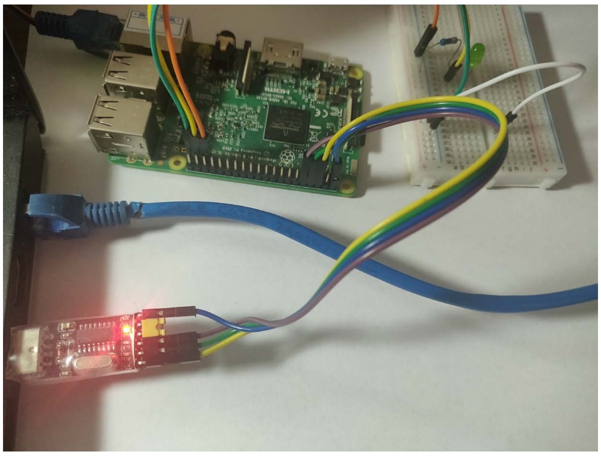


Figure 5. Experimental hardware equipment connection diagram.

Table 1. Equipment configuration table.

Equipment	System/Software	IP, MAC address
Server system	Linux aarch64	4.15.0-70-generic
Raspberry Pi System	Raspbian GNU/Linux 10	Raspberry3+
Virtual machine 1 (Linux system)	Ubuntu 14.04.4	processor: 1 RAM: 2GB
Virtual machine 2 (Linux system)	Linux kali 4.19.0	processor: 1 RAM: 2GB
Virtual machine software	VMware® Workstation 15 Pro	15.0.2 build-10952284
Pfsense		2.4.5

Table 2. System ladder diagram settings.

Variable Name	Type	Position
control	BOOL	%IX1.5
lamp	BOOL	%IX1.6

Figure 6 is the corresponding ladder diagram of the experimental system, by converting the system into a ladder diagram.

Import the .st file generated by the OpenPLC editor into OpenPLC to monitor the status of the control in real time. As shown in Figure 7, Monitoring is a function of OpenPLC, and the refresh time is set to 100 milliseconds.

Use Python to write programs to achieve Modbus communication. The server serves as the host and the Raspberry Pi serves as the slave. Use WireShark to capture Modbus communication data. The captured data is shown in Figure 8.

4.2. DoS Attacks on the System

The system attack experiment uses Kali Linux, which is a Debian-based Linux system distribution, mainly used for digital forensics, penetration testing, and hacker defense. In Kali, we attacked the Raspberry Pi system. The attack code is as follows:

```
Root@kali: ~#fping-asg 192.168.0.0/24;
Root@kali: ~#hping3-q-n-a 1.1.1.1--icmp-d 200--flood 192.168.1.119;
```

After Kali executes a denial of service attack, Snort will detect it, and it will send an alert when it detects an attack.

4.3. DoS Attack on Modbus Protocol

Attacks on Modbus use the SMOD framework, which is an open source industrial control system Modbus communication attack framework. Using this framework, any system that uses Modbus communication can be attacked. The attack methods of the SMOD framework include obtaining device function codes, ARP attacks on devices, and DoS attacks on devices.

As shown in Figure 9, using SMOD to obtain the UID, RHOTS, and PORT of the device, the device can be attacked.

4.4. Attack Detection

Pfsense is based on FreesBSD and can be installed on a computer as a firewall and router in the network. This experiment mainly uses its firewall function.

Snort is an open source network intrusion detection system. It can not only be used for rule matching, but also can detect anomalies, cross-packet attacks, and protocol non-standard attacks.

One advantage of Pfsense is that it can add a variety of plug-ins. After adding the Snort plug-in, it has the function of intrusion prevention system. Pfsense can monitor LAN and WAN. This implementation uses its monitoring of WAN and combines snort rules to alert on attacks, making the virtual chemical control system based on OpenPLC more secure.

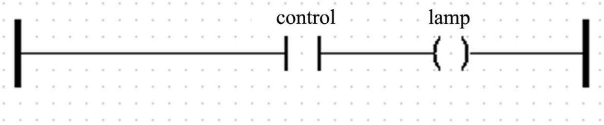
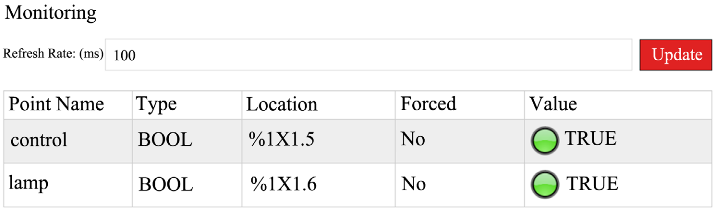


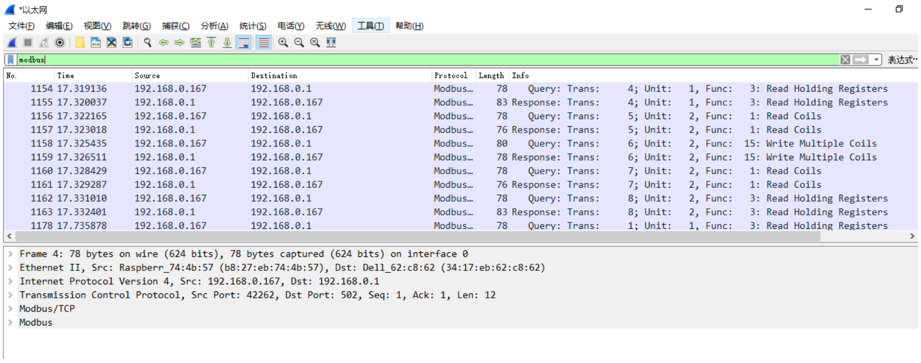
Figure 6. System ladder diagram.


Monitoring

Refresh Rate: (ms) 100

Point Name	Type	Location	Forced	Value
control	BOOL	%1X1.5	No	● TRUE
lamp	BOOL	%1X1.6	No	● TRUE

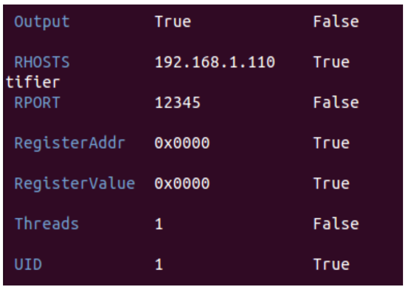
Figure 7. OpenPLC monitoring situation.


*以太网
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
表达式…

```
modbus
No.    Time          Source          Destination     Protocol  Length  Info
  1154 17.319136     192.168.0.167   192.168.0.1     Modbus…   78     Query: Trans:    4; Unit:   1, Func:   3: Read Holding Registers
  1155 17.320037     192.168.0.1     192.168.0.167   Modbus…   83 Response: Trans:    4; Unit:   1, Func:   3: Read Holding Registers
  1156 17.322165     192.168.0.167   192.168.0.1     Modbus…   78     Query: Trans:    5; Unit:   2, Func:   1: Read Coils
  1157 17.323018     192.168.0.1     192.168.0.167   Modbus…   76 Response: Trans:    5; Unit:   2, Func:   1: Read Coils
  1158 17.325435     192.168.0.167   192.168.0.1     Modbus…   80     Query: Trans:    6; Unit:   2, Func:  15: Write Multiple Coils
  1159 17.326511     192.168.0.1     192.168.0.167   Modbus…   78 Response: Trans:    6; Unit:   2, Func:  15: Write Multiple Coils
  1160 17.328429     192.168.0.167   192.168.0.1     Modbus…   78     Query: Trans:    7; Unit:   2, Func:   1: Read Coils
  1161 17.329287     192.168.0.1     192.168.0.167   Modbus…   76 Response: Trans:    7; Unit:   2, Func:   1: Read Coils
  1162 17.331010     192.168.0.167   192.168.0.1     Modbus…   78     Query: Trans:    8; Unit:   2, Func:   3: Read Holding Registers
  1163 17.332401     192.168.0.1     192.168.0.167   Modbus…   83 Response: Trans:    8; Unit:   2, Func:   3: Read Holding Registers
  1178 17.735878     192.168.0.167   192.168.0.1     Modbus…   78     Query: Trans:    1; Unit:   1, Func:   3: Read Holding Registers

> Frame 4: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
> Ethernet II, Src: Raspberr_74:4b:57 (b8:27:eb:74:4b:57), Dst: Dell_62:c8:62 (34:17:eb:62:c8:62)
> Internet Protocol Version 4, Src: 192.168.0.167, Dst: 192.168.0.1
> Transmission Control Protocol, Src Port: 42262, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
> Modbus/TCP
> Modbus
```

Figure 8. Modbus data captured.



```
 Output         True            False

 RHOSTS         192.168.1.110   True
tifier
 RPORT          12345           False

 RegisterAddr   0x0000          True

 RegisterValue  0x0000          True

 Threads        1               False

 UID            1               True
```

Figure 9. SMOD attack.

Based on the Snort engine, design rules for matching, that is, the rule language. The rules used to detect DoS in this experiment are as follows:

{alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;}

In this rule, alert means that an alarm will be issued if the entry rule is triggered, icmp means protocol type, any means any source/destination IP address, any/80 means port number, -> means direction operator, msg means in alarm And the message issued in the package log, sid represents the ID of the Snort rule, and rev is used to identify the version of the rule modification.

In addition, it also includes rules for specific communication protocols such as Modbus communication protocol. The rules used to detect Modbus TCP function code scanning attacks in this experiment are as follows:

```
{
alert tcp $192.168.0.1 502 -> $192.168.1.107 any
(flow: established;
content: “|00 00|”; offset:2;
depth:2; byte_test: 1, >=, 0x80, 7;
content: “|01|”offset:8; depth:1;
msg: “SCADA_IDS: Modbus TCP-Function Code Scan”;
threshold: type threshold, track by_src, count 3, seconds 60;
classtype: attempted-recon; sid:1111014; rev:2; priority:2;)
}
```

In this rule, content means searching for the specified pattern in the payload of the data packet, offset means the offset of the match, and depth means the depth of the match. The above constitutes the Snort rules for exclusive Modbus communication.

The intrusion detection test needs to be processed by the Snort engine. Through the analysis of the data packet, it matches with the data packet signature. If it succeeds, an early warning is generated.

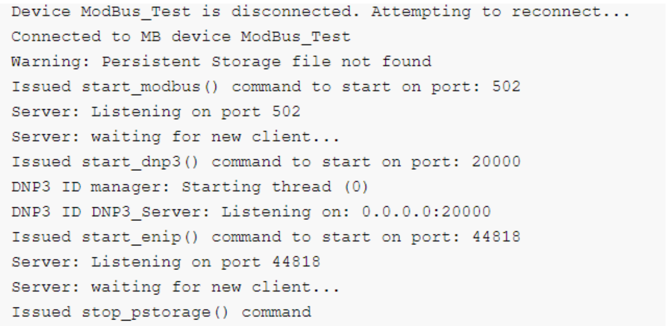
5. Result Analysis

5.1. Result and Analysis of DoS Attacks on the System

After OpenPLC is successfully deployed in the system and started, a successful connection message will be reported in the log. It can be seen from the log information that OpenPLC will try to connect to the added slave station, and execute the start_Modbus command on port 502, and monitor port 502 (Figure 10).

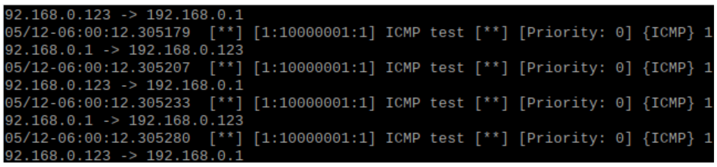
After using Kali to attack, you can obviously feel that the running speed of Raspberry Pi is slowing down, and Snort will send out an alarm. Snort detection results are shown in Figure 11.

It can be seen from Figure 11 that this attack uses the ICMP protocol, using the ping command to perform a denial of service attack on the Raspberry Pi’s IP address 192.168.0.123. It can be concluded from the experimental results that snort can detect attacks well.



```
Device ModBus_Test is disconnected. Attempting to reconnect...
Connected to MB device ModBus_Test
Warning: Persistent Storage file not found
Issued start_modbus() command to start on port: 502
Server: Listening on port 502
Server: waiting for new client...
Issued start_dnp3() command to start on port: 20000
DNP3 ID manager: Starting thread (0)
DNP3 ID DNP3_Server: Listening on: 0.0.0.0:20000
Issued start_enip() command to start on port: 44818
Server: Listening on port 44818
Server: waiting for new client...
Issued stop_pstorage() command
```

Figure 10. OpenPLC log.



```
92.168.0.123 -> 192.168.0.1
05/12-06:00:12.305179  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 1
92.168.0.1 -> 192.168.0.123
05/12-06:00:12.305207  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 1
92.168.0.123 -> 192.168.0.1
05/12-06:00:12.305233  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 1
92.168.0.1 -> 192.168.0.123
05/12-06:00:12.305280  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP} 1
92.168.0.123 -> 192.168.0.1
```

Figure 11. Snort alarm information.

After the Raspberry Pi is subjected to a denial of service attack, the remote desktop connection will be disconnected. As shown in Figure 12, the OpenPLC log will also report an error.

5.2. Result and Analysis of Modbus Attack Experiment

The attack destination IP is 192.168.1.105, so first use the SMOD framework to obtain the UID of the attacker for subsequent attacks. The process of obtaining UID is as follows:

```
SMOD modbus(uid) > set RHOSTS 192.168.1.105
SMOD modbus(uid) > exploit
[+] Module Brute Force UID Start
[+] Start Brute Force UID on: 192.168.1.105
[+] UID on 192.168.1.105 is: 1
```

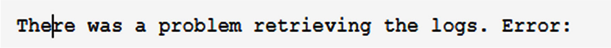
It can be seen from the above operations that the UID of the device is 1, and then the SMOD framework can be used to further attack the device. The process is as follows:

```
SMOD modbus(writeSingleRegister) > set RHOSTS 192.168.1.105
SMOD modbus(writeSingleRegister) > set RegisterValue 0x0000
SMOD modbus(writeSingleRegister) > set RegisterAddr 0x0000
SMOD modbus(writeSingleRegister) > set UID 1
SMOD modbus(writeSingleRegister) > exploit
```

By using the write to a single register in the SMOD attack, set the IP of the attacker RHOTS, set the value of the register to be written to 0x0000, set the address of the register to be written to 0x0000, and set the UID to 1. The result is as follows:

```
[+] Module Write Single Register Start
[+] Connecting to 192.168.1.105
[+] Response is:
###[ModbusADU]###
transId = 0x7
protoId = 0x0
len = 0x6
unitId = 0x1
###[Write Single Register Answer]###
funcCode = 0x6
registerAddr = 0x0
registerValue = 0x1
```

Use wireshark to capture this attack process, and you can find this packet with



```
There was a problem retrieving the logs. Error:
```

Figure 12. OpenPLC log error.

function code 06. The captured data packet is shown in **Figure 13**, which is the attack process of SMOD writing a single register.

By attacking a single register, the value of the register can be changed. From the attack process, it can be known that the value of the register whose address is 0x0000 is changed from 0x0001 to 0x0000. In this experiment, the address is the register on 0x0000 It controls the on and off of the LED light, 0x0001 is high level, that is, the LED light is on, and becomes low level 0x0000 after being attacked, that is, the LED light is off. In the digital virtualization system built, it can be seen that before the attack, Value is TRUE, and after the attack, Value is FALSE, thus realizing the monitoring of the industrial control system by the digital virtualization system (**Figure 14**).

Through the pfsense firewall, the flow of the attack process can be observed. Set the WAN interface monitored by pfsense and display it through the IP address. You can see the traffic changes detected by pfsense under different attack methods, as shown in **Figure 15**.

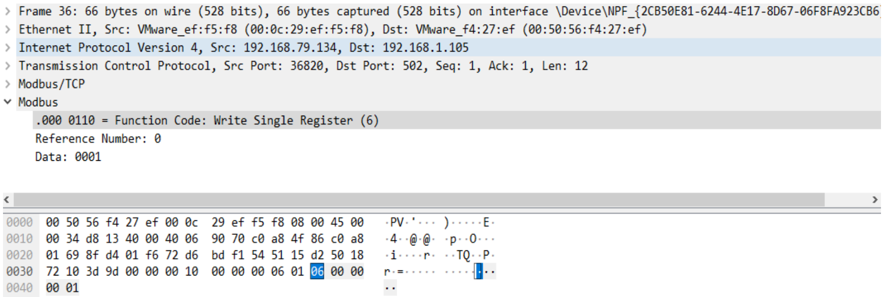
In **Figure 15(a)** is the flow of the client and server under normal traffic conditions, and only a few fluctuations can be seen; **Figure 15(b)** is a DoS attack on the system that writes a single register, and the fluctuations can be seen The amplitude is large and dense, with the maximum speed reaching 250 k Bits/sec or more; **Figure 15(c)** is the attack of writing a single register. It can be seen that the traffic reaches 5 k Bits/sec, which shows a short flow change; **Figure 15(d)** is writing For the DoS attacks on all registers, because the registers are written into the attack one by one, it can be seen that the traffic fluctuates. It is obvious that the interval between before and after the attack has reached a maximum speed of 200 k Bits/sec.

Figure 16 is a system monitoring diagram, and its parameters are user utilization (user util.), priority utilization (nice util.), system utilization (system util.), interrupt (interrupt) and processes (processes). As can be seen from the figure, when the system is subjected to a DoS attack, the system utilization rate (system util.) increases significantly, with the highest value reaching 90.21%, which means that a large amount of system resources will be consumed during the DoS attack, resulting in a high system utilization rate.

After completing this attack process, visit the snort homepage in pfsense and view the alert interface to view the alert information of this attack.

As shown in **Figure 17**, after being attacked, the words "Someone abnormally connected to the modbus device" will be displayed, and information about the attack event, protocol used, source IP, target IP, and attacked port will be displayed.

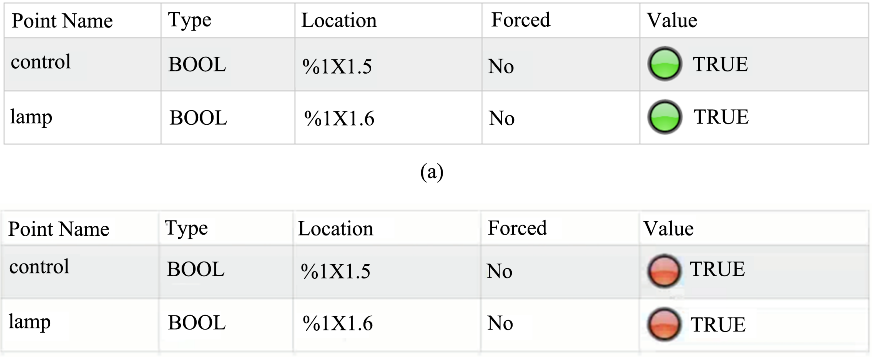
After attacking the Raspberry Pi slave side, snort will issue an alarm and block the destination IP and source IP. You can see snort's blocking information from **Figure 18**. It can be found that the attack initiated by the attacker was successfully detected and prevented.



```
> Frame 36: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{2CB50E81-6244-4E17-8D67-06F8FA923CB6
> Ethernet II, Src: VMware_ef:f5:f8 (00:0c:29:ef:f5:f8), Dst: VMware_f4:27:ef (00:50:56:f4:27:ef)
> Internet Protocol Version 4, Src: 192.168.79.134, Dst: 192.168.1.105
> Transmission Control Protocol, Src Port: 36820, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
> Modbus/TCP
v Modbus
    .000 0110 = Function Code: Write Single Register (6)
    Reference Number: 0
    Data: 0001

0000  00 50 56 f4 27 ef 00 0c  29 ef f5 f8 08 00 45 00   ·PV·'··· )·····E·
0010  00 34 d8 13 40 00 40 06  90 70 c0 a8 4f 86 c0 a8   ·4··@·@· ·p··O···
0020  01 69 8f d4 01 f6 72 d6  bd f1 54 51 15 d2 50 18   ·i····r· ··TQ··P·
0030  72 10 3d 9d 00 00 00 10  00 00 00 06 01 06 00 00   r·=····· ········
0040  00 01                                              ··
```

Figure 13. Capture network packets during the attack.



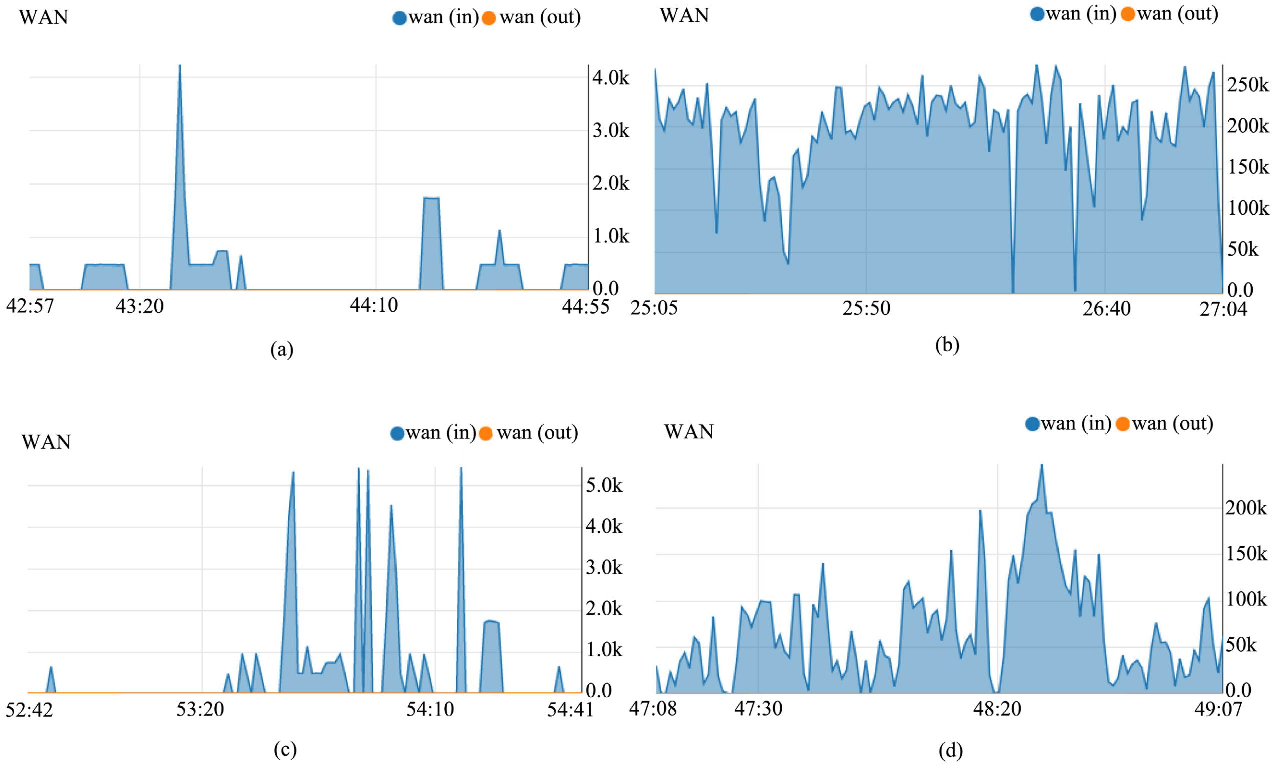
Point Name	Type	Location	Forced	Value
control	BOOL	%1X1.5	No	🟢 TRUE
lamp	BOOL	%1X1.6	No	🟢 TRUE

(a)

Point Name	Type	Location	Forced	Value
control	BOOL	%1X1.5	No	🔴 TRUE
lamp	BOOL	%1X1.6	No	🔴 TRUE

(b)

Figure 14. State changes of the OpenPLC before and after attack. (a) Pre-attack state; (b) Post-attack state.



(a)

(b)

(c)

(d)

Figure 15. Traffic changes under different attacks. (a) Original state; (b) DoS write single register; (c) Write a single register; (d) DoS write all registers.

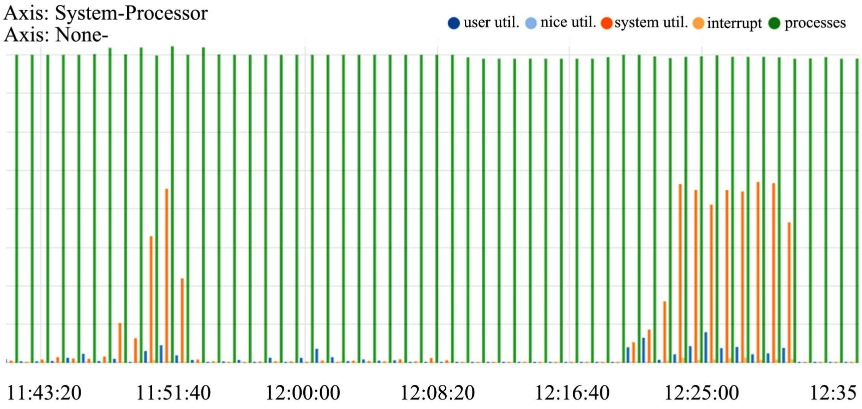
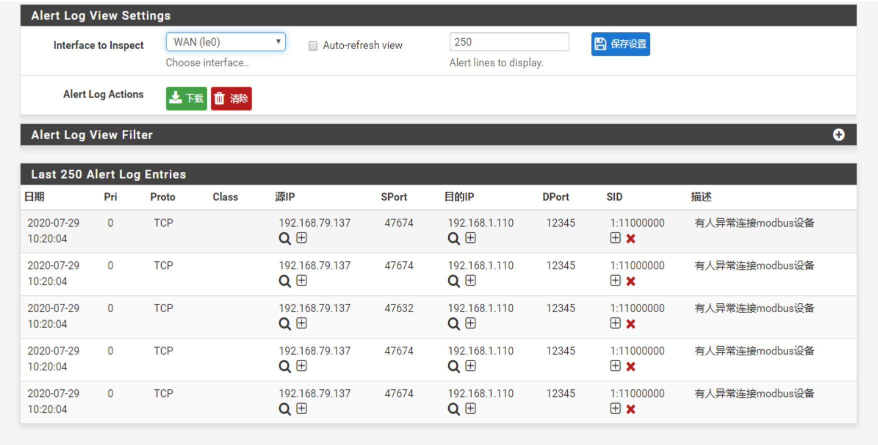


Figure 16. System monitoring.



Alert Log View Settings

Interface to Inspect: WAN (le0) — Choose interface..
□ Auto-refresh view
250 — Alert lines to display.
保存设置

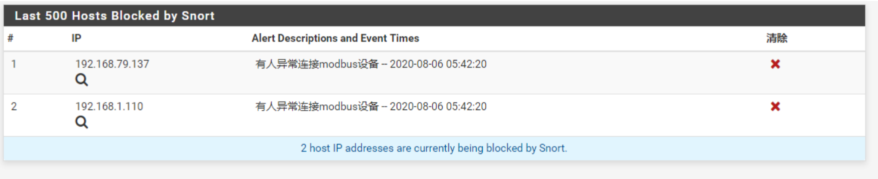
Alert Log Actions: 下载 清除

Alert Log View Filter

Last 250 Alert Log Entries

日期	Pri	Proto	Class	源IP	SPort	目的IP	DPort	SID	描述
2020-07-29 10:20:04	0	TCP		192.168.79.137	47674	192.168.1.110	12345	1:11000000	有人异常连接modbus设备
2020-07-29 10:20:04	0	TCP		192.168.79.137	47674	192.168.1.110	12345	1:11000000	有人异常连接modbus设备
2020-07-29 10:20:04	0	TCP		192.168.79.137	47632	192.168.1.110	12345	1:11000000	有人异常连接modbus设备
2020-07-29 10:20:04	0	TCP		192.168.79.137	47674	192.168.1.110	12345	1:11000000	有人异常连接modbus设备
2020-07-29 10:20:04	0	TCP		192.168.79.137	47674	192.168.1.110	12345	1:11000000	有人异常连接modbus设备

Figure 17. Snort issued a warning.



Last 500 Hosts Blocked by Snort

#	IP	Alert Descriptions and Event Times	清除
1	192.168.79.137	有人异常连接modbus设备 – 2020-08-06 05:42:20	✖
2	192.168.1.110	有人异常连接modbus设备 – 2020-08-06 05:42:20	✖

2 host IP addresses are currently being blocked by Snort.

Figure 18. Snort blocking information.

6. Conclusions

The industrial control virtualization system is at risk of being attacked. After being attacked, the industrial control system will be disconnected from the control, and the main console cannot detect the operating status of the system, or the data has been tampered with, resulting in wrong operations, and thus cannot be timely Maintenance will cause a lot of losses.

In view of the continuous expansion of industrial control systems and the gradual emergence of network security issues, this paper proposes a digital virtual industrial control system, builds a virtual industrial control system based on OpenPLC, and attacks on this system, using Snort intrusion detection The pfsense firewall of the system detects Dos attacks. The construction of a virtual industrial control system can monitor and predict the actual industrial control system, simulate possible attacks, and design a plan to deal with the attack. Therefore, the attack detection security scheme proposed in this paper is useful for the future development and security of industrial control systems. It has practical significance. In addition, this method also has limitations, that is, there is no smarter method to improve system security, and so in-depth research will be conducted in this area in the future.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Gartner (2017) Gartner's Top 10 Strategic Technology Trends for 2017[EB/OL]. https://www.forbes.com/sites/peterhigh/2016/10/18/gartner-top-10-strategic-technology-trends-for-2017/

[2] Panetta, K. (2018) 5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018. https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/

[3] Panetta, K. (2019) 5 Trends Appear on the Gartner Hype Cycle for Emerging Technologies, 2019. https://www.gartner.com/smarterwithgartner/5-trends-appear-on-the-gartner-hype-cycle-for-emerging-technologies-2019/

[4] Panetta, K. (2020) 5 Trends Drive the Gartner Hype Cycle for Emerging Technologies, 2020. https://www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020/

[5] Tao, F., Liu W., Liu J.H., *et al.* (2018) Digital Twin and Its Potential Application Exploration. *Computer Intergrated Manufacturing Systems*, **24**, 1-18.

[6] Alves, T.R., Buratto, M., de Souza, F.M., *et al.* (2014) OpenPLC: An Open Source Alternative to Automation. *IEEE Global Humanitarian Technology Conference* (*GHTC* 2014), San Jose, CA, USA, 10-13 October 2014, 585-589. https://doi.org/10.1109/GHTC.2014.6970342

[7] Ma, Y., Wang C.-X., Yin Z.-Y., *et al.* (2019) Design and Construction of Digital Twin Laboratory Model Based on SAMP of Chinese Academy of Science. *Computer Intergrated Manufacturing Systems*, **40**, 2343-2347.

[8] Tao, F., Liu, W.R., Zhang, M., *et al.* (2019) Five-Dimension Digital Twin Model and Its Ten Applications. *Computer Intergrated Manufacturing Systems*, **25**, 1-18.

[9] Tao, F., Zhang, M. and Cheng, J.F. (2017) Digital Twin Workshop: A New Paradigm for Future Workshop. *Computer Intergrated Manufacturing Systems*, **23**: 1-9.

[10] Chen, M., Deng, C.-Y., Zhang, J. and Guo, R.-F. (2020) Research on 3D Detection and Interaction Algorithm of Production Line Based on Digital Twin. *Journal of Chinese Computer Systems*, **41**, 979-984.

[11] Liu, Z.F., Chen, W., Yang, C.B., Cheng, Q. and Zhao, Y.S. (2019) Intelligent Manufacturing Workshop Dispatching Cloud Platform Based on Digital Twins. *Computer Intergrated Manufacturing Systems*, **25**, 1444-1453.

[12] Wang, S.L., Wang, Y.K., Yang, B. and Wang, S.B. (2019) Fog Manufacturing: New Paradigm of Industrial Internet Manufacturing Based on Hierarchical Digital Twin. *Computer Intergrated Manufacturing Systems*, **25**, 3070-3080.

[13] Xiong, Q., Liu, H., Xu, Y., *et al.* (2015) A Vulnerability Detecting Method for Modbus-TCP Based on Smart Fuzzing Mechanism. 2015 *IEEE International Conference on Electro/Information Technology* (*EIT*), Dekalb, IL, USA, 21-23 May 2015, 404-409. https://doi.org/10.1109/EIT.2015.7293376

[14] Chen, B., Pattanaik, N., Goulart, A., *et al.* (2015) Implementing Attacks for Modbus/TCP Protocol in a Real-Time Cyber Physical System Test Bed. 2015 *IEEE International Workshop Technical Committee on Communications Quality and Reliability* (*CQR*), Charleston, SC, USA, 11-14 May 2015, 1-6. https://doi.org/10.1109/CQR.2015.7129084

[15] Li, X., Liu, X. and Wan, X.X. (2019) Overview of Digital Twins Application and Safe Development. *Journal of System Simulation*, **31**, 385-392.