# Study on the Evidence Collection for Network Security Intrusion Detection

Xindong Wang
*(Corresponding author: Xindong Wang)*
Shaanxi Police College, Xi'an, Shaanxi, China
No. 199, Qiyuan Second Road, Weiyang District, Xi'an, Shaanxi 710021, China
Email: muxin090786@163.com
*(Received May 23, 2019; Revised and Accepted Aug. 13, 2022; First Online Oct. 15, 2022)*

## Abstract

The expansion of the Internet not only brings convenience but also increases the risk of network information security. This paper briefly introduced the software-defined network (SDN) technology and the SDN-based network intrusion detection and evidence collection system. The system had a dynamic priority scheduling strategy for intrusion detection and evidence collection response tasks. The designed network intrusion detection and evidence collection system were simulated in a small SDN built in a laboratory and compared with the other system adopting a static priority task scheduling strategy. The results suggested that the designed network intrusion detection and evidence collection system effectively collected suspicious traffic; compared with the network intrusion detection and evidence collection system using the static priority task scheduling strategy, the designed detection and evidence collection system was comparable in terms of detection accuracy but had more advantages in terms of evidence collection time and switch transmission rate.

*Keywords: Evidence Collection; Intrusion Detection; Network Security; Software-Defined Network*

## 1 Introduction

With the globalization of information and the rapid development of the Internet, companies, governments, and individuals are cooperating on the Internet to enhance work efficiency and convenience and enrich people's leisure time [1]. Networks have two advantages, open and sharing. On the one hand, the two advantages give the network various resources, facilitating people's life; on the other hand, cyber criminals can also take advantage to launch an invasion against enterprises, governments, or individuals through the network [2].

In recent years, the popularity of mobile terminals that can access the Internet has also diversified the ways of cybercrime, which poses greater security risks to users than the convenience it brings. Therefore, in order to protect users' network security, it is necessary to make effective protection against intrusion data and record them in the process of protection to facilitate the traceability and forensics of the intrusion data [3]. Lu *et al.* [1] proposed a new model based on the optimized back-propagation neural network (BPNN) and Dempster-Shafer theory to detect intrusion data and verified the effectiveness of the method through experimental simulations.

Jing *et al.* [5] proposed a network intrusion detection method based on associative deep learning and found from simulation results that the method had a high average detection rate and average error detection rate for unknown intrusions and attacks. Xie *et al.* [6] proposed an online distributed intrusion detection model based on a cellular neural network and found from the experiment on the KDD CUP 99 dataset that the method was feasible and effective. This paper briefly introduced the software-defined network (SDN) technology and the SDN-based network intrusion detection and evidence collection system, simulated the designed system in a small SDN built in a laboratory, and compared it with an intrusion detection and evidence collection system adopting another task scheduling strategy.

## 2 Detection and Evidence Collection of Network Intrusion Data

### 2.1 Introduction of SDN Technology

The expansion of the Internet has greatly facilitated people's lives, but it has also provided a corresponding platform for law-breakers. The Internet's development brings not only convenience but also network security problems. For example, law-breakers will use the vulnerability of Internet protocols to hijack normal users' computers as "broiler chickens" and launch distributed denial of service (DDoS) attacks consisting of a large amount of malicious traffic to target users [7]. In the face of increasing network security problems, the artificial extension of con-

figurations for network security devices is difficult due to the complex structural system of the traditional network, making the efficient automatic response difficult [8]. SDN technology has emerged as a new network security protection strategy.

Compared with the traditional network structure, SDN has only three structures: application layer, control layer, and data layer, which is relatively simpler in network maintenance. In SDN, data transmission and control are separated, i.e., data are transmitted using SDN switches in the data layer, but the switches are only responsible for the transmission of data and do not have the function of deciding the destination of data transmission. The function of controlling data transmission is realized in the control layer. The control layer receives tasks from the application layer and sends corresponding control commands to the switch in the data layer, and the switch opens and closes ports according to the commands [9].

Since SDN separates control and transmission, it is much easier to extend the network. The switch in the data layer is only responsible for data transmission, so it is not necessary to consider the structural changes of the data layer when extending the SDN, but only to add the corresponding extended interfaces in the control layer and the corresponding interactive interface software in the application layer.

## 2.2 SDN-based Network Intrusion Detection and Evidence Collection

As introduced in the previous section on SDN, the network structure has good scalability after separating the control and transmission of data, but even if the network structure is optimized, malicious attacks on the Internet will not be reduced. Due to the characteristics of the SDN structure, DDoS is a common form of attack [10]. During a DDoS attack, a large amount of false or meaningless traffic data is used to make the SDN control generate a large number of requests and flow entries, thus taking up the computing resources of the controller and switches and eventually bringing the network down.

In order to resist DDoS attacks, SDN needs to conduct the corresponding detection in the process of data transmission to distinguish normal data from abnormal data and then intercept the abnormal data. At the same time, the data judged to be abnormal are collected and stored, which can be used as evidence for recourse or a template to further improve the identification performance of the abnormal data detection algorithm [11].

Figure 1 shows the basic structure of the SDN-based network intrusion detection and evidence collection system. Since the network intrusion detection and evidence collection system is based on SDN, its basic structure is also divided into three layers. The application layer contains a DDoS interception function and an evidence collection function. These two functions in the application layer have a user-oriented interface so that users can intuitively see the DDoS interception results and alerts and

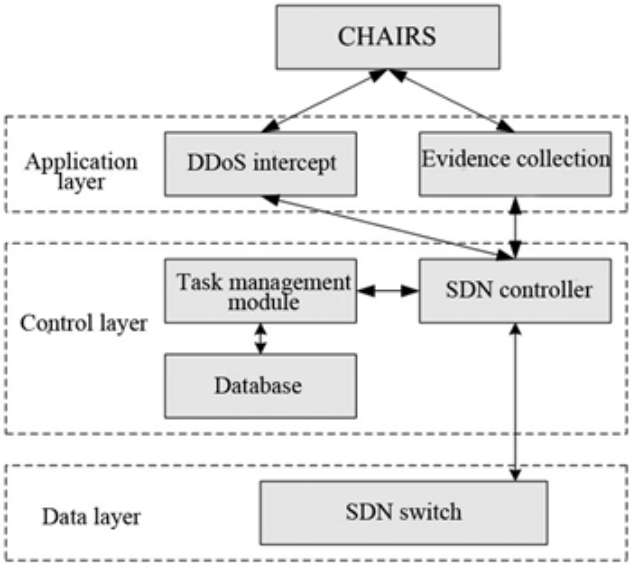


Figure 1: Basic structure of the SDN-based network intrusion detection and evidence collection system

the evidence collection report of abnormal data [12]. The full name of "CHAIRS" in Figure 1 is cooperative hybrid aided incidence response system, which is a distributed network emergency management system that mainly receives alerts from the active security defense system in SDN and responds to them. It is a distributed network emergency management system that receives and responds to alerts from the active security defense system in the SDN.

The control layer includes the SDN controller, task management module, and database. The SDN controller is responsible for receiving requests from the application layer, converting them into flow entries, and sending them to the switch at the data layer. It also plays the role of switch monitoring in the process of network intrusion detection and evidence collection [13]. The database not only caches the task requests but also stores the suspicious messages obtained from the evidence collection for storage. The task management module is responsible for managing the tasks issued by the application layer and allocating computing resources according to the task priority.

The data layer is composed of SDN switches, whose role is to transmit data and follow the flow entries issued by the SDN controller when transmitting data [14]. Data transmission in SDN is entirely the responsibility of the switch. The control layer can control the direction of data transmission by simply controlling the opening and closing of switch ports. In other words, the data traffic transmitted in the SDN structured network does not directly pass through the control and application layers, which makes the security of the control and application layer devices improved, but DDoS does not directly damage the control and application layer devices but uses junk

traffic to occupy computing resources, which is considered an attack method for the characteristics of SDN structure. Therefore, SDN networks still need to make active protection against DDoS intrusion attacks and collect evidence.

### 2.3 The Process of Network Intrusion Detection and Evidence Collection

The network intrusion detection and evidence collection process based on the SDN structure is shown in Figure 2. In this process, the SDN controller needs to perform tasks including opening and closing switch ports and collecting evidence of suspicious traffic transmitted by the switch. Every task is quite heavy, but the computational resources of the whole system are limited, so it is impossible to process all tasks together. Therefore, it is necessary to follow the scheduling policy to enable the tasks to be executed. The traditional scheduling strategy assigns different priority fields to different tasks and executes the tasks in the order of priority, but this fixed priority scheduling strategy will lead to serious polarization of computing resources between low priority and high priority tasks, so this paper adopts a dynamic priority scheduling strategy to allocate queues [15], and the specific steps are shown in Figure 2.

1) The server in the SDN controller responsible for monitoring the switch traffic collects the feature fields of the traffic data in the switch.

2) The collected traffic feature field is uploaded to the DDoS interception module in the application layer. The DDoS attack is detected using the corresponding DDoS detection algorithm. It returns to Step 1 if no DDoS attack is detected; if a DDoS attack is detected, an early warning is sent to CHAIRS through the system interface.

3) The CHAIRS receives the alert and issues response tasks to the DDoS interception module and evidence collection module in the application layer through the system interface according to the set script.

4) After receiving the response tasks, the DDoS interception and evidence collection modules in the application layer both generate the corresponding control commands and send the task commands to the SDN controller through the application programming interface (API).

5) After receiving the task commands, the SDN controller judges whether the current computing resources of the system can support the task command according to the scheduling policy of the task management module. If it can, the task command is put into the execution queue; if not, it is put into the waiting queue.

6) The task command is selected from the waiting queue in priority order. Whether the remaining system resources can support the execution of the task command is determined. If it can, the task command is put into the execution queue. If not, whether the task can be replaced by a lower priority task in the execution queue to obtain computer resources is determined. If it can, the two tasks are exchanged; if not, the task is put back into the waiting queue after adding one to its priority. It is recorded as one scheduling cycle when Step 6 is cycled once.

7) In every scheduling cycle, the SDN controller receives tasks from the execution queue in priority order, generates the corresponding flow entries, and sends them to the SDN switch through the OpenFlow protocol.

8) The SDN switch controls the data transmission based on the received flow entries, including the closing or opening of the attacked ports and the transmission of messages of suspicious traffic to the corresponding database. Finally, it returns to Step 1.

## 3 Simulation Experiments

### 3.1 Experimental Setup

The simulation experiment was conducted in a small SDN built in a laboratory. Figure 3 shows the basic architecture of the SDN for the simulation experiment. The whole network intrusion detection and evidence collection system had four servers and one switch. One server was used as the SDN controller, which took the role of the application and control layers, and one server was used as the database for storing messages of suspicious traffic collected. The remaining two servers served as regular servers for simulating two users who transmitted data. The switch played the role of data transmission. The switch was connected to all the servers, but the flow entry information was exchanged between the SDN controller and the switch through the OpenFlow protocol. The switch was used as a transit for data interaction between the regular servers, and it was a one-way connection from the switch to the database.

In the whole process of network intrusion detection and evidence collection, the transmission traffic of the switch was monitored using the switch monitoring module in the SDN controller, and the traffic data was detected using the intrusion detection algorithm. A BPNN was used to warn the traffic data. When the intrusion data were detected, a warning was issued. After receiving the warning, the CHAIRS created response tasks, and then the task management module in the SDN controller scheduled and assigned the tasks, converted them to flow entries in order, and sent them to the switch. The switch transmitted the data according to the flow entries, including intercepting the malicious data and collecting them as evidence.

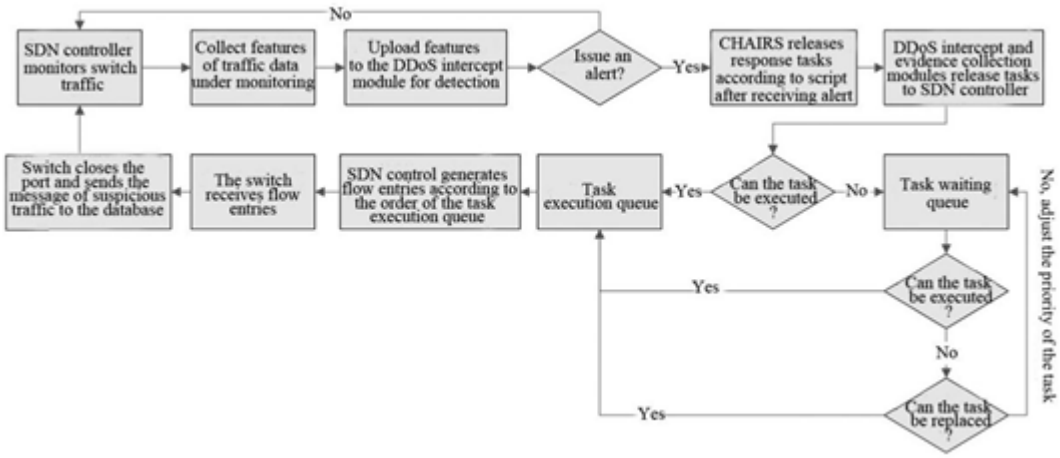


Figure 2: The basic process of network intrusion detection and evidence collection

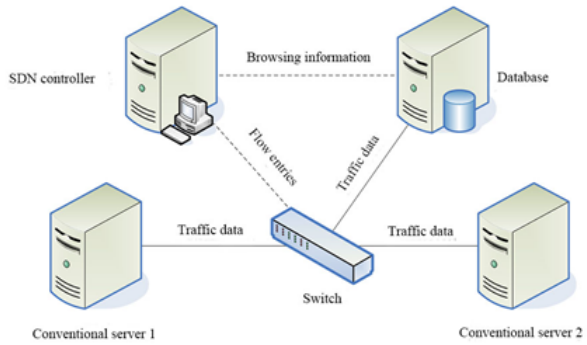


Figure 3: Architecture diagram of the SDN-based network intrusion detection and evidence collection system

## 3.2 Experimental Projects

Two thousand packets of data of different sizes were sent from conventional server 1 to conventional server 2, and 500 were treated as anomalous packets, i.e., intrusion data. The designed network intrusion detection and evidence collection system was used to detect packets and collect evidence. In order to further verify the effectiveness of the improved task scheduling strategy, a network intrusion detection forensics system adopting the conventional static priority task scheduling strategy was also simulated for comparison. The system used for comparison was architecturally consistent with Figure 3, but the only difference was the task scheduling strategy used by the task management module within the SDN controller.

## 3.3 Experimental Results

The designed network intrusion detection and evidence collection method was used in the simulation experiment to detect and collect evidence from the data transmitted between conventional servers 1 and 2. As the amount of data acting as abnormal data was large during the experiment, the number of response tasks generated during the experiment was also very large. Limited by space, only the feedback result of evidence collection response task number 3 is shown here, as shown in Figure 4. It was seen from the feedback result that the evidence collection task number of this feedback result was 3, which started at 18:42:41 on June 13, 2022, and ended at 18:55:36 on June 13, 2022, and 1.26 MB of suspicious messages were successfully collected, which triggered 365 alerts in the transmission process.

To verify the performance of the proposed network intrusion detection and evidence collection method, it was compared with a network intrusion detection and evidence collection system with the same structure but a different task scheduling strategy. Table 1 shows the performance test results of network intrusion detection and evidence collection under two task scheduling strategies. The P value in the comparison of the detection accuracy between the traditional static priority task scheduling strategy and the dynamic priority task scheduling strategy was 0.165, i.e., the difference was not significant. The P value of the comparison of the average time of evidence collection was 0.01, and the method adopting the dynamic priority task scheduling strategy was faster in collecting suspicious data. The P value in the comparison of the average data transmission rate in the switch was 0.01, and the detection and evidence collection method that adopted the dynamic priority task scheduling strategy had higher switch data transmission rate.

The reason for these results was analyzed. The two network intrusion detection and evidence collection methods only differed in their task scheduling strategies. For the network intrusion detection and evidence collection method, the accuracy of network intrusion data detection depended on the intrusion detection algorithm. Both methods used a BPNN to detect traffic data in the simulation experiment, so they were comparable in terms of detection accuracy. In terms of the evidence collec-

Table 1: Network intrusion detection forensics performance under two task scheduling policies

| | Detection accuracy/% | The average time spent on evidence collection/s | The average data transmission rate of the switch MB/s |
|---|---|---|---|
| Traditional static priority task scheduling strategy | 98.6 | 265 | 523 |
| Dynamic priority task scheduling strategy | 98.5 | 203 | 869 |
| P value | 0.165 | 0.01 | 0.01 |

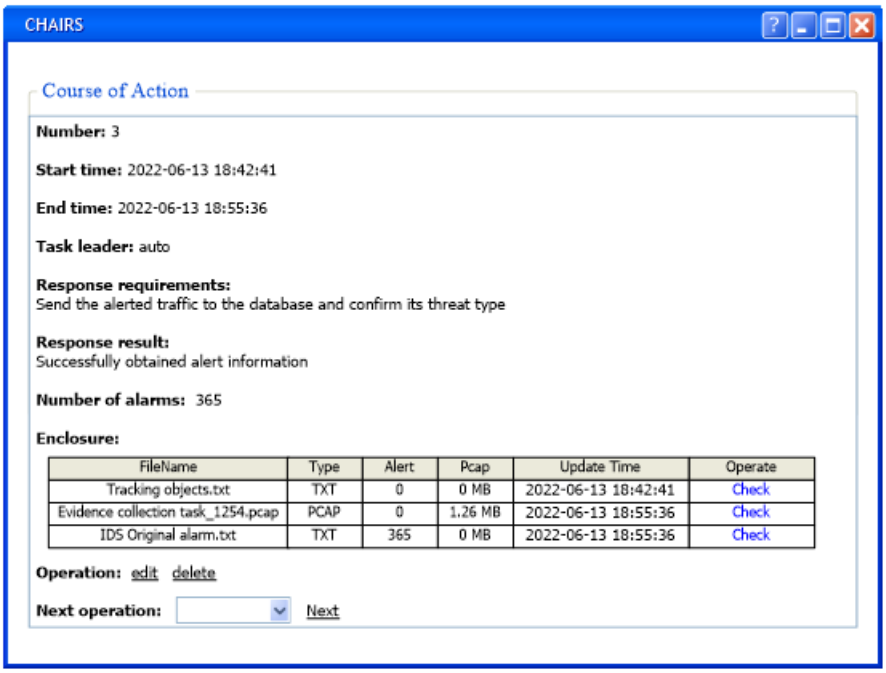


Figure 4: The feedback result of forensic task number 3

tion time and switch data transmission rate, the method that adopted the dynamic priority task scheduling strategy was more advantageous. The reason is as follows. The traditional static priority task scheduling strategy processed tasks in priority order, but if the new response task had a high priority for a long time, tasks with low priorities in the waiting queue would not be treated for a long time, i.e., new tasks with high priorities would cut in line, affecting the actual task processing. The dynamic priority task scheduling policy processed tasks in priority order as a whole, but low-priority tasks tried to replace with lower-priority tasks in the execution queue, and if that did not work, they would adjust their priorities upward to be selected in the execution queue in the next scheduling cycle.

## 4 Conclusion

This paper briefly introduced the SDN technology and the SDN-based network intrusion detection and evidence collection system, which had a dynamic priority scheduling strategy for intrusion detection and evidence collection response tasks. The proposed network intrusion detection and evidence collection system was simulated in a small SDN built in a laboratory, and it was compared with the network intrusion detection and evidence collection system adopting a static priority task scheduling strategy. The following findings were obtained. The network intrusion detection and evidence collection system could effectively collect suspicious traffic in the network. The intrusion detection and evidence collection systems adopting different task scheduling strategies were not significantly different in the detection accuracy of suspicious data, but the intrusion detection and evidence collection system adopting the dynamic priority scheduling strategy was more advantageous in the evidence collection time and switch transmission rate.

## References


[1] S. Khan, A. Gani, A. Wahab, M. Shiraz, I. Ahmad, "Network forensics: Review, taxonomy, and open challenges," *Journal of Network & Computer Applications*, vol. 66, no. May, pp. 214-235, 2016.

[2] C. Jain, A. K. Saxena, "General study of mobile agent based intrusion detection system (IDS)," *Journal of Computer and Communications*, vol. 4, no. 4, pp. 93-98, 2016.

[3] G. Xu, "Research on network intrusion detection method based on machine learning," *Journal of Physics: Conference Series*, vol. 1861, no. 1, pp. 1-6, 2021.

[4] C. Lu, L. Yue, M. Ma, N. Li, "A hybrid NIDS model using artificial neural network and D-S evidence," *International Journal of Digital Crime & Forensics*, vol. 8, no. 1, pp. 37-50, 2016.

[5] L. Jing, W. Bin, "Network intrusion detection method based on relevance deep learning," in *International Conference on Intelligent Transportation*, pp. 237-240, 2016.

[6] K. Xie, Y. Yang, Y. Xin, G. Xia, "Cellular neural network-based methods for distributed network intrusion detection," *Mathematical Problems in Engineering*, vol. 2015, no. pt.3, pp. 1-10, 2015.

[7] G. Fanani, I. Riadi, "Analysis of digital evidence on denial of service (DoS) attack log based," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 2, no. 2, pp. 70, 2020.

[8] S. Kalime, "Efficient network intrusion detection system using Boyer Moore algorithm index termsnetwork intrusion detection system; packet capturing module, Boyer-Moore," *International Journal of Research*, vol. 4, no. 17, pp. 1083, 2017.

[9] F. Shang, D. Zhou, C. Li, H. Ye, Y. Zhao, "Research on the intrusion detection model based on improved cumulative summation and evidence theory for wireless sensor network," *Photonic Network Communications*, vol. 37, no. 2, pp. 1-12, 2018.

[10] Z. Wang, "Network intrusion detection by using combination optimizing features and classifier parameters," *Journal of Nanjing University of Science & Technology*, vol. 41, no. 1, pp. 59-64, 2017.

[11] D. Gugelmann, F. Gasser, B. Ager, V. Lenders, "Hviz: HTTP(S) traffic aggregation and visualization for network forensics," *Digital Investigation*, vol. 12, pp. S1-S11, 2015.

[12] L. Duan, F. Yu, L. Zhan, "An improved fuzzy C-means clustering algorithm," in *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 44-46, 2016.

[13] P. Nayak, A. Devulapalli, "A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime," *EEE Sensors Journal*, vol. 16, no. 1, pp. 137-144, 2015.

[14] T. Yoshioka, S. Karita, T. Nakatani, "Far-field speech recognition using CNN-DNN-HMM with convolution in time," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4360-4364, 2015.

[15] N. Khamphakdee, N. Benjamas, S. Saiyod, "Improving intrusion detection system based on Snort rules for network probe attacks detection with association rules technique of data mining," *Journal of ICT Research & Applications*, vol. 8, no. 3, pp. 234-250, 2015.


## Biography

**Wang Xindong** was born in Hanzhong City, Shaanxi Province. He obtained a master's degree from Chang'an University in June 2011. He is an engineer and is working at Shaanxi Police College. He is interested in the network security of industrial control systems.