An Intrusion Prevention Scheme for Malicious Network Traffic Based on SDN

Xiaofeng Xu¹, Jiahao Dai²,∗ and Yifang Zhi³
¹,²Science and Technology on Communication Information Security Control Laboratory, China
¹,²China Electronic Group Corporation NO.36 Institute, Zhejiang Jiaxing, China
³Xidian University, Shanxi Xi’an, China

∗Corresponding author

Abstract. In recent years, SDN technology has developed rapidly, and the security of SDN is the key to its further development and application. Intrusion prevention system possesses both the features of intrusion detection and protection, which is one of the important methods to ensure the security of SDN network. In this paper, we proposed a SDN-based intrusion prevention scheme for SDN security. The scheme utilizes the programmability of SDN to create four modules on the application plane of the controller, including the network monitoring module, IP address detection module, destructive packets detection and Snort linkage module. The scheme also provides users with Web UI and manual/automatic operation mode. Finally, we deployed the scheme on the Mininet platform with the Floodlight controller.

1. Introduction
Software defined network (SDN) is a new type of network architecture that separates the control layer and forwarding layer of the network. This separation makes the underlying network infrastructure abstract and unified with respect to network applications and services, so that it can be directly programmed. SDN can accelerate the deployment of new services and network innovation [10]. However, due to the centralized management, openness and programmability of SDN, it also brings a certain security threat to SDN: Due to the centralized control, the network will fall into chaos, once the attacker successfully implements the attack on the controller.[8]. The security issues of SDN have also attracted widespread attention from industry, enterprise organizations, research groups and standardization organizations. A considerable amount of literature has been published on the security of SDN [1].

In the protection of the network, the intrusion prevention system (IPS) compensates for the defect of the intrusion detection system (IDS) that cannot defend attacks actively, and also compensates for the defect of the firewall that cannot detect external traffic. The IPS for traditional networks has also been extensively explored by predecessors. Li [3] deeply discusses the basic working principle of IPS, and expounds the relationship between IPS and IDS, and proposes a prediction that IPS should be better developed on the basis of IDS. Wan [2] proposes a network-based Intrusion Prevention System (NIPS), which uses embedded methods to implement the deployment of the IPS, and combines honeypot technology to further analyze the network for behavior only, Chen [4] establishes a Snort-based IPS by transplanting the Snort core detection engine. The above intrusion prevention systems are all applied in the traditional network environment. The way to deploy to the network is tandem. It is necessary to
capture network traffic in real time and detect the packets before forwarding them, which will have a certain impact on the operating efficiency of the network. Compared with the traditional network architecture, the network architecture of SDN is quite different. Therefore, the traditional IPS cannot be directly applied to SDN. It needs to be combined with the separation of the data layer and the control layer to establish a new type of intrusion prevention system. Wei [5] designs the Formatting the title, authors and affiliations Snort linkage module on the Ryu controller to realize the linkage between Ryu controller and Snort, and establishes an IPS based on snort. Zhang [7] proposes a scheme for deploying IDS on a virtual switch. This scheme dynamically blocks intrusion by establishing an IDS IP table form, but this aggregate the control layer and the data layer, thus violating the principle of SDN.

Based on the above references, this paper carefully compares the strengths and weaknesses of each intrusion prevention scheme and proposes an SDN-based intrusion prevention scheme for malicious network traffic. Using the SDN programmable features, we developed the network monitoring module, IP address detection module and destruction message monitoring module on the application layer, which realizes the function of network monitoring and data filtering of the southbound interface. Based on these modules, we established the Snort linkage module to realize the information interaction between the controller and the Snort agent deployed on the switch. Finally, we provide users with Web UI.

2. Background

2.1. SDN
SDN originated from an internal research project at Stanford University in the United States, which was proposed to solve various problems in the traditional network architecture. Once the SDN technology was put forward, it has received widespread attention from academia and industry. SDN has also become one of the more researches in next-generation network technologies [3]. The SDN architecture decouples the traditional closed network system into the application layer, the control layer and the data layer, and logically realizes the centralized control and management of the network.

The SDN architecture decouples the traditional closed network system into the application layer, the control layer and the data layer, logically realizes the centralized control and management of the network, and further provides a good platform for improving the development and innovation of network applications [9]. Its architecture diagram is shown in Figure 1.

![SDN Architecture](image)

**Figure 1.** SDN architecture.

The data layer is mainly composed of various switches in the network, and these devices only have a forwarding function. The most typical switch in the data layer is a virtual (OpenvSwitch, OVS) switch. Each switch has flow tables, and the switch performs corresponding actions on the corresponding data packet according to the flow table. The control layer is the core part of the SDN architecture. The SDN
controller works on this layer. The controller interacts with the network device of the data layer through the southbound interface. The most typical southbound interface is the OpenFlow protocol. The control layer controls the entire data and holds the information of the network topology of the data layer. SDN controllers can generate flow rules and can send them to each switch of the data layer [6]. The application layer is composed of various network applications that implement control of the network based on the northbound interface provided by the control layer to meet specific needs.

2.2. IDS and IPS
IDS is an important means to protect network security. It can be divided into host-based, network-based and hybrid IDS according to the deployment locations. According to the different detection methods of the analysis engine, it can be divided into abnormal detection and misuse detection [5]. In the intrusion detection system, Snort has a good reputation in the industry and is powerful enough to detect multiple network attacks. Snort is an open source software written in C that conforms to the GNU General Public License (GPL) specification. Snort is a network-based misuse intrusion detection system. It uses libpcap software to capture packets on the network, and analyzes packets with malicious data matching. After suspicious data is detected, snort will issue an alert and record log. Because snort supports multiple platforms, its structure is clear and its code is open source, it becomes a model for researching IDS/IPS.

IPS is a proactive defense system deployed on critical paths of the network to detect network traffic through itself. When a threatening traffic is detected, it sends an alert message and actively filters out the traffic. IPS is equivalent to the combination of IDS and firewall. It both has the IDS detection function and the firewall filtering function.

3. SDN-based Intrusion Prevention Scheme for Malicious Network Traffic

3.1. Overall Design
Programmability is one of the core features of SDN. Based on this feature, applications for various needs can be developed at the application layer of SDN. The intrusion prevention scheme designed in this paper is mainly deployed on the application layer of SDN. The scheme is implemented by developed new application modules combined with the embedded application module on the Floodlight controller. The overall architecture of the system is shown in Figure 2.

![Figure 2. Intrusion prevention system overall architecture.](image-url)
As shown in Figure 3, the solution is specifically divided into two working modes: a manual mode and
an automatic mode. The manual mode is mainly used when a network administrator exists, and the
network administrator processes various warning information. In automatic mode, the controller
automatically handles various threats. The entire solution integrates an IP address detection module, a
destruction packet detection module, a network monitoring module, and a Snort linkage module on the
Floodlight controller.

1) The main function of the IP address detection module is to bind the IP address, MAC address, and
physical location information of each network device, thereby preventing IP spoofing attacks and
tracking the attack source.

2) The main function of the destruction packet detection module is to analyze the flag bits of the data
packet received by the controller, detect the corrupted data packet and discard it. Thereby preventing
further propagation of data packets with destructive properties in the network.

3) The main function of the network monitoring module is to monitor the network speed, link delay
and the status of the switch port, so as to realize the perception of the entire network situation and
timely discover the dangerous situation and malicious traffic (eg. DDOS 、 Tor 、 APT) in the network.

4) The main function of the Snort linkage module is to realize the communication and linkage
between the controller and the Snort agent deployed on the OVS switch. The Snort agent can transmit
a warning message to the controller to further process the threat traffic.

In addition, this solution provides users with web UI, on which users can view network monitoring
information and control the monitoring module. In the manual mode, each warning message generated
by each module is also transmitted to the web UI for processing by the user.

3.2. IP Address Detection Module

In the SDN architecture, the control layer is separated from the data layer. The introduction of flow
table technology makes the relationship between the addresses of different network devices in
different levels in the traditional network more compact. Combined with this feature, we designed the
IP address detection module. The implementation idea of this module is to bind the IP address, MAC
address and physical location of each network device. The binding information of the network device
can provide reference for IP address detection. The implementation process of the module is shown in
Figure 3.

Figure 3. Flow chart for implementing the spoofing detection function.
3.3. Destruction Message Detection Module
The detection object of this module is a Packet-in message. When the switch fails to match the flow entry in the process of forwarding the data packet, the switch encapsulates the data packet into a Packet-in message and sends it to the controller. This module is used to detect the data packets generated in this case. The detection of data packets that can be matched to the flow table entries is mainly done by the Snort agent. The function of this module is shown in Figure 4. The main process is to classify the data packets and then detect and filter the data packets according to different standards.

![Flow chart for destroying the message detection function.](image)

3.4. Network Monitoring Module
The network monitoring module mainly includes three functions: rate measurement, delay measurement, and port status collection. The implementation idea of this module is that the controller sends various types of request messages and special Packet-out messages to the switch, and analyzes the messages returned by the switch and the Packet-in messages to obtain corresponding monitoring data. The overall architecture of the module is shown in Figure 5.

![Overall architecture diagram of the network monitoring module.](image)

As shown in Figure 5, the rate measurement function mainly relies on the Flow-Status-Request message sent by the controller and the Flow-Status-Reply message returned by the switch. The delay measurement function is mainly implemented by constructing a special Packet-out message, listening and analyzing the Packet-in message returned by the switch for this special Packet-out message. The port status collection function mainly relies on the Port-Status-Request message sent by the controller and the Port-Status-Reply message returned by the switch.
3.5. Snort Linkage Module
The module is mainly composed of three parts: the Snort linkage module on the controller side, the daemon process on the switch side, and the Snort agent. On the switch side, the Snort agent will be deployed to a port on the switch. It will detect the traffic passing through the port according to Snort's rule repository. When abnormal traffic is detected, an alert will be generated and the message will be delivered to the switch. The daemon acts as a relay and passes this alert message to the controller. The main function of the Snort linkage module on the controller side is to receive the alert message sent from the switch. The overall architecture of the module is shown in Figure 6. The thick black solid line is the information flow process in the manual mode, the virtual black line is the flow of information in the automatic mode, and the thin black solid line is the information flow process common to the two modes.

![Figure 6 Snort linkage module overall architecture diagram.](image)

4. The Implementation of the Scheme

4.1. Experimental Environment
To test the feasibility and functionality of the intrusion prevention solution, we need to set up a test environment. The version information of each component of the test environment is shown in Table 1.

| Table 1. Test platform components selection. |
|---------------------------------------------|
| operating system | Ubuntu   |
| controller       | Floodlight |
| switch           | Openvswitch |
| topology         | Mininet  |
| snort            | Snort    |

In order to test the system's protection against intrusion, we need to use some network attack tools to simulate the attack behavior. We use the cyber attack tool shown in Table 2 to simulate cyber attack behavior.

| Table 2. Network attack tool. |
|------------------------------|
| ARP spoofing tool | Ettercap |
| Port scan tool     | Nmap    |
| Packet tool        | Sendip  |
| Dictionary generation tool | Crunch |
| Violent cracking tool | Hydra |

Create a test topology using Mininet. The topology is shown in Figure 7.

4.2. Feasibility Verification
After the topology is created, we use the network attack tool mentioned above to attack the network to verify the feasibility of the intrusion prevention solution.
4.2.1. Flood attack protection. Open the xterm terminal of the attacking host and enter the ping -f command to perform an ICMP flood attack on Web Server 1. Then, enable the network monitoring module. When the module finds that the bandwidth utilization has exceeded the threshold, the module judges it as a flood attack and adds an ACL filtering rule to prevent the attack. Open the access control list, we can find that ACL rules have been added.

![Topology](image)

**Figure 7.** Topology.

4.2.2. ARP man-in-the-middle attack protection. Open the xterm terminal of the attacking host and use ettercap to launch the ARP man-in-the-middle attack. At this point, the module will detect the ARP man-in-the-middle attack and discard the corresponding ARP spoofing packet. The module then generates an alert message and sends it to the web UI.

4.2.3. Nmap scan protection. Open the xterm terminal of the attacking host and use the nmap tool to perform a FIN scan of Web Server 2. At this point, Snort Agent 2 detects the FIN scanning attack on Web Server 2 by the attacking host. It will generate an alert message, and then pass the alert message to the controller through the daemon process. After receiving the alert message, the controller will send this alert message to the web UI for user processing. The module can also invoke the information binding table of the IP address detection module to quickly trace the attack source.

4.2.4. Violent attack protection. Open the xterm terminal of the attacker and use the hydra brute force tool to brute force the username and password of the FTP server. During the brute force attack, the Snort agent detects the attack and generates an alert message and sends it to the alert event page for the user to process. In automatic mode, firewall rules will be added automatically.

4.3. Performance Evaluation

In order to compare the performance difference between the scheme and the Iptable-based IPS scheme, we deployed the Iptable-based IPS in the same environment and installed the IDS Informer software for sending attack data to generate attack traffic in the virtual topology and simulate the real cyber attacks. We then generate attack packets and normal data packets in the virtual topology at different rates, and take the method of averaging multiple tests (10 tests respectively) to test the intrusion detection efficiency of the traditional IPS solution and the SDN-based IPS solution. The result is shown in Figure 8.

As shown in Figure 8, as the attack rate increases, the efficiency of the intrusion detection of the traditional IPS scheme decreases first. When the attack rate is 12000 packets/s, the detection efficiency decreases to 85%. The SDN-based IPS scheme has a detection efficiency of more than 90% at an attack rate of 40,000 packets/s. Compared with the traditional IPS solution, the intrusion data processing capability of the SDN-based intrusion prevention scheme is improved by more than 2 times.
5. Conclusion

The application and promotion of SDN technology will change the traditional network architecture, control traffic from the entire network layer, improve the deployment speed of new services, save operating costs, and provide a new solution for network virtualization technology. However, the security problem of SDN is the key problem that restricts the further development and application of SDN technology. This paper proposes a SDN-based intrusion prevention scheme for the security problem in SDN environment and implements a lot of experiments to prove the effectiveness of the scheme. But we have not studied the SDN-based intrusion prevention algorithm. We leave exploration of this venue to future work.

References

[1] Shuling Wang, Jihan Li, Yunyong Zhang, et al. SDN Architecture and Security Research[J]. Telecommunications Science, 2013, 29(3): 117-122.
[2] Minmin Wan. Research on Intrusion Detection Technology Based on User Anomaly Level in Virtual Machine Environment [D]. Guangxi University, 2013.
[3] Lihui Li. Research on Intrusion Prevention System[J]. Computer Security, 2008(9): 68-70.
[4] Yong Chen. Research on performance optimization of intrusion prevention system based on Snort[D]. University of Electronic Science and Technology of China, 2014.
[5] Shujun Wei. Design and implementation of network security isolation system based on SDN technology [D]. Beijing University of Posts and Telecommunications, 2015.
[6] Wenting Ma. Research on Key Technologies of SDN Controller Based on OpenFlow [D]. Beijing University of Posts and Telecommunications, 2015.
[7] Xiang Zhang. Design and implementation of SDN-based traffic awareness and bandwidth control system [D]. University of Electronic Science and Technology of China, 2015.
[8] Yoon C, Lee S, Kang H, et al. Flow Wars: Systemizing the Attack Surface and Defenses in Software-Defined Networks[J]. IEEE/ACM Transactions on Networking, 2017, PP(99):1-17.
[9] Bosshart P, Gibb G, Kim H S, et al. Forwarding metamorphosis: fast programmable match-action processing in hardware for SDN[J]. Computer Communication Review, 2013, 43(4):99-110.
[10] Zuo Q Y, Chen M, Zhao G S, et al. Research on OpenFlow-Based SDN Technologies[J]. Journal of Software, 2013, 24(5):1078-1097.