The security function dynamic calling scheme for software-defined security

Yuanyuan Ma¹, Zhimin Guo²*, Wen Yang²

¹ Global Energy Interconnection Research Institute, Nanjing 210003, China
² Henan Electric Power Research Institute, Zhengzhou 450052, China
*Corresponding author’s e-mail: zhiming.guo@163.com

Abstract. With the rapid development of cloud computing, various network attack methods are becoming more advanced, and the attack speed is getting faster. Traditional security protection solutions can no longer adapt to new network attacks, bringing significant challenges to security protection in virtualized network environments such as cloud computing. Based on these problems, this paper proposes a dynamic calling mechanism of security functions for software-defined security based on software-defined network technology. This mechanism defined the strategies of dynamically scheduling of virtual security service nodes in the virtualized network environment. The strategies describe the network's data flow and virtual security devices by extending the attribute-based access control policy model. It can dynamically construct the mapping relationship between network flow and virtual security devices according to users' security requirements, thus forming a personalized security service chain for specific network flows. This mechanism can protect the "north-south" network flow at the IoT edge agent according to the user security business requirements.

1. Introduction
With the advent of the mobile office era, the traditional data center based on server unified management began to transform into a large-scale cloud computing and cloud storage center. The security service chain of traditional network pulls the network data flow meeting specific attributes through the service sequence composed of multiple business function service nodes to provide malicious attack protection. However, the traditional network service chain is closely combined with the physical network topology, and hardware security devices are serially connected to the network flow path through manual configuration of various protection strategies, which leads to large coupling degree between devices, serious topology dependence and rigid security protection scope, which makes it difficult to meet the requirements of rapid business iteration in cloud computing environment.

The emergence of SDN, Software Defined Networking) technology has laid the foundation for realizing the fine-grained access control strategy based on "data flow". In SDN network based on OpenFlow protocol, the forwarding operation of network flow no longer depends on the forwarding tables of layer 2 and layer 3 of traditional network elements, but is completed by the controller of control plane and the switch of data forwarding layer, and the controller and switch communicate through the southbound protocol OpenFlow protocol [2]. SDN architecture based on OpenFlow protocol has good programmability and openness, which can conveniently achieve the goal of automatically and flexibly pulling network data flow through multiple security business function service nodes for detection and protection. The architecture decouples the control plane and data plane of network security devices. The bottom layer virtualizes the traditional hardware security devices and pools resources through Network
Function Virtualization (NFV) technology to form a unified security resource pool. The top layer defines and describes the network data flow in a fine-grained manner by software definition, and redirects the target network data flow to the security service sequence composed of multiple virtual security devices for security protection, to realize flexible scheduling of network flow and flexible deployment of security devices [3]. Therefore, the software-defined security architecture provides a new idea for building a fast, flexible and automated security protection mechanism in the cloud environment.

The work of this paper mainly solves the following problems: firstly, how to build a security service chain policy for users by extending the attribute-based access control policy model, and arrange virtual security devices according to priority to solve policy conflicts; Secondly, how to realize the virtual security device instance with the smallest real-time load to join the security service chain is solved to avoid the single point failure of the security device; finally, it studies how to implement the routing engine mechanism by extending SDN controller to accomplish traffic redirection. The main contributions of this paper are as follows:

• Solve the policy conflict problem caused by fine-grained access control and complex network flow attributes, and study how to ensure the correct implementation of user policies to meet the diverse security business needs of users. Through research, the effective arrangement method of virtual security devices is put forward to ensure the correct implementation of the policy.
• To balance the load of the same type of virtual security devices and solve the optimization problem of security resource scheduling, this paper studies how to complete the evaluation of security resource scheduling based on the real-time load of virtual security devices. Through research, this paper puts forward a security resource scheduling mechanism and algorithm which can make full use of the load (CPU utilization rate, memory utilization rate) of virtual security equipment and avoid single point failure of security equipment.
• To solve the problem of network flow redirection, the path calculation and flow table issuing mechanism of SDN controller are analysed, and the flow table automatic generation and update mechanism of network flow redirection is designed, to realize the function of pulling network flow for security detection in turn according to the arrangement order of virtual security devices.

In this paper, the Minijet network simulation tool and Linux image are used to build a virtualized network environment, and build a secure resource pool to complete the test of the security service chain automatic arrangement and deployment framework. It is vital that you do not add any headers, footers or page numbers to your paper; these will be added during the production process at IOP Publishing (this is why the Header and Footer margins are set to 0 cm in table 1).

2. Overall Framework
Software Defined Security (SDS) refers to the technical idea of separating control and forwarding from SDN, decoupling traditional hardware security devices into standard universal hardware and security software functions, and opening programmable interfaces at the upper layer to realize automatic and flexible arrangement and management. The bottom layer is abstracted as a centrally managed security resource pool composed of virtualized security devices, which realizes unified registration, pool management and flexible deployment of security resources.

The main work of this paper is to propose a dynamic calling mechanism of security functions for software-defined security. This mechanism needs to automatically generate security service chain policies, dynamically arrange virtual security devices, and quickly schedule network data flow to specific virtual security device sequences to achieve the goal of security protection according to the security business requirements of upper-level users. The mechanism consists of three parts: the security service chain policy conflict decision node, the security resource scheduling node and the SDN controller. The policy conflict decision node is responsible for building a security service chain policy for users' security needs and making policy conflict decisions; The security resource scheduling node is responsible for managing the resource usage of virtual security devices in the security resource pool and selecting suitable virtual security device instances for the security service chain strategy; SDN controller
manages and monitors network topology from a global perspective, and realizes network flow redirection function according to policies. The overall architecture is shown in Figure 1.

Figure 1 An automatic scheduling framework of security service chain based on SDN/NFV.

3. Security Service Chain

3.1. Description of Security Service Chain Policy
The description of security requirements is the basis and premise of automatic arrangement and deployment of security service chain. Attribute-based access control model (ABAC) provides dynamic permission allocation and more fine-grained access control in complex information systems, with good flexibility and scalability [14]. To describe the network flows and virtual security devices in SDN environment, the mapping relationship between them is constructed, and the personalized security service chain for specific network flows is formed. This paper describes the user's security business requirements by extending the ABAC policy model, and then constructs the security service chain policy. Each security business requirement of users can be described as a policy, and each policy can build a security service chain. The subject, object, action authority and VSA priority of virtual security equipment can be shaped.

The policy syntax tree is shown in Figure 2.

Figure 2 Policy syntax tree

3.2. Policy Conflict Decision Algorithm
Security service chain policy constructs the mapping relationship between network flow and VSA sequence of virtual security devices. To ensure the correct implementation of all security service chain policies and make network flow correctly protected by VSA sequence of virtual security devices corresponding to security service chain policies, it is necessary to establish a complete policy conflict decision-making mechanism. Therefore, priority is adopted to dynamically and reasonably arrange the
VSA sequences of virtual security devices in the policy action authority of the security service chain to solve the policy conflicts and meet the diverse security business requirements of users.

Policy conflict decision algorithm: after the policy conflict decision node constructs the security service chain policy for the user's security business requirements, firstly, the SDN controller retrieves the current flow rules including the network flow corresponding to the 12 tuples in the newly constructed security service chain policy, and deletes them to trigger the Packet-in message. Then, the policy conflict decision node listens to all Packet-in messages received by the controller, reconstructs corresponding security service chain policies for all network flows described by these messages in turn, and updates the implementer of the policies as the security resource scheduling node (the policies are reconstructed from the security service chain policies issued by users, and then deleted after being processed and converted into flow rules in turn by the security resource scheduling node and SDN controller), so as to solve the policy conflict problem.

3.3. Secure resource scheduling algorithm
The default Forwarding module of SDN controller listens to the Packet-in message and calculates a forwarding route from A to B based on the source and destination address information of the packet. To realize the traffic redirection function, that is, to drag the network flow from A to B to C, this paper extends the controller to design the routing engine mechanism to realize the traffic redirection operation. The routing engine receives and parses the security service chain policy sent by the security resource scheduling node, generates flow rules in turn and sends them to the switch for traffic redirection operation, and then deletes the security service chain policy. In this paper, all the flow rules sent by the controller are stored to facilitate the execution of the flow rule deletion operation.

3.3.1. Secure The construction of security service chain
Security service chain is described by policies, and each policy can be constructed as a security service chain. For example, when defending against DDoS attacks, users need to build a policy to schedule network flows to pass through Web protection equipment WAF and traffic cleaning equipment ADS in turn to form fine-grained security protection. The security service chain corresponding to this policy can be formally expressed as follows:

$$ServiceChain = [(Src, D1, D2, Dst)] | D1 \in D, D2 \in D$$  \hspace{1cm} (1)

In the actual network flow forwarding route, it can be divided into 1, 1, 2, 2 and 3 forwarding routes. Then yes, there are:

$$Path = [PathSrc, D1, PathD1, D2, ..., PathDn, Dst]$$ \hspace{1cm} (2)

Among them, it is the network flow forwarding route of the current security service chain, and forwarding routes are generated according to the VSA priority of virtual security devices from big to small, and forwarding routes exist between adjacent virtual security devices.

3.3.2. Secure The construction of security service chain
The routing engine receives the security service chain policy. First, it extracts the source host src_ip, destination host dst_ip and all security device instances D1, D2, D3, ..., Dn, information to provide parameter information for routing calculation. Routing engine generates network flow forwarding routes in the following two situations:

- If the number of VSA sequences of the virtual security device in the security service chain policy is 1. Firstly, the network flow forwarding route from the source host src_ip to the flow entrance of the VSA instance of the virtual security device is calculated; Then calculate the network flow forwarding route from the traffic outlet of the virtual security device VSA instance to the destination host.
- If the number of VSA sequences of virtual security devices in the security service chain policy is greater than 1. Firstly, the forwarding routes inside the VSA sequences of virtual security devices are calculated in order of priority from big to small, such as Path(D1,D2), Path(D2,D3) etc. Then, the forwarding route from the source host to the first instance (i.e., the instance with the highest priority) in
the VSA sequence of the virtual security device is calculated. Finally, calculate the forwarding route from the traffic outlet of the last VSA instance (that is, the instance with the lowest priority) to the destination host \(\text{dst}_{i,p}\).

4. EXPERIMENT AND EVALUATION

4.1. Experimental environment

To more reasonably verify the effectiveness and flexibility of the security protection of the SDN/NFV-based security service chain automatic arrangement and deployment framework in the virtualized environment, a virtualized network environment hosted on Intel Xeon 2.00GHz 132GB RAM physical server is built, and its network topology is shown in the following figure 3. The network topology has a good representative significance. On the one hand, the "north-south" and "east-west" traffic communication in the cloud environment is abstracted as the communication among hosts H1, H2 and H3, and OVS1 is used as the border switch for accessing the cloud environment as a secure resource pool, thus simplifying the complex network structure of OpenStack cloud computing environment. On the other hand, the security resource pool constructed by eight OpenFlow switches and four virtual security devices is also convenient to verify the security service chain automatic arrangement and deployment framework proposed in this paper.

4.2. Analysis of results

In the experiment, both hosts H1 and H2 keep pinging host H3, and at the same time, tcp packets with port 81 are continuously sent to host H3 through the network performance testing tool Iperf at a certain request rate, and then the above four security service chain policies are issued in turn through the northbound interface of the security service chain framework. Among them, Mininet has its own Iperf tool, and Iperf client can continuously send udp and tcp packets to the server.

- During initialization, when there is no security service chain policy in the network, after receiving these network flow requests, the switch OVS1 encapsulates their first data packet as a Packet-in message and requests the controller to establish dynamic flow rules. Therefore, hosts H1, H2 and H3 communicate normally. To better present the forwarding route of network flow, the actual forwarding route of network flow is presented in the form of schematic diagram below, the result of port packet capture of host H3 is shown in Figure 4.

![Figure 3 Network topology of experiment](image-url)
First, policy 1 is issued through the northbound interface of the security service chain. Since the forwarding flow rules of host H1's ping and tcp requested network flow are all included in the object domain of policy 1, these flow rules are deleted first, and then the security service chain strategy is reconstructed for host H1's ping and tcp network flow, which is first pulled to FW and then sent to host H3. The forwarding route of host H2 remains unchanged. The FW entry port packet capture record is shown in Figure 5.

5. Conclusion
With the rapid development of cloud computing, various network attack methods are becoming more advanced and the attack speed is getting faster. Traditional security protection solutions can no longer adapt to new network attacks, which brings great challenges to security protection in virtualized network environments such as cloud computing. In view of the above problems, this paper proposes a dynamic calling mechanism of security functions for software-defined security based on software-defined network technology. This mechanism studies the strategy of dynamically arranging virtual security service nodes in virtualized network environment, describes the data flow and virtual security devices in the network by extending the attribute-based access control policy model, and dynamically constructs the mapping relationship between network flow and virtual security devices according to users' security requirements, thus forming a personalized security service chain for specific network flows. Aiming at the "north-south" and "east-west" security protection problems of cloud platform, this mechanism is verified and tested by experiments in the process of automatic arrangement and deployment of security service chain framework in virtualized environment. It is verified that this mechanism can realize "north-south" network flow protection at the edge IoT agent, and realize flexible and rapid deployment according to user security business requirements.

Acknowledgments
Supported by the science and technology project of State Grid Corporation of China: "Research on The Security Protection Technology for Internal and External Boundary of State Grid information network Based on Software Defined Security" (No. 5700-202058191A-0-0-00);

References
[1] Lv, Zhihan, Wenqun Xiu. "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT. " IEEE Internet of Things Journal 7.7 (2019): 5706-5712 . J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[2] Alsaeedi, Mohammed, Mohd Murtadha Mohamad, Anas A. Al-Roubaiey. "Toward adaptive and scalable OpenFlow-SDN flow control: A survey. " IEEE Access 7 (2019): 107346-107379.
[3] Herrera, Juliver Gil, and Juan Felipe Botero. "Resource allocation in NFV: A comprehensive survey." IEEE Transactions on Network and Service Management 13.3 (2016): 518-532.

[4] Zheng J, Li Q, Gu G, et al. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(7): 1838-1853.

[5] Fang D, Yuxiang H U, Ou L I. Routing framework and creation algorithm in Ad Hoc based SDN. Journal on Communications, 2019, 40(9): 33.

[6] Yubo S, Huiwen Y, Wei W U, et al. Joint DDoS detection system based on software-defined networking. Journal of Tsinghua University (Science and Technology), 2019, 59(1): 28-35.

[7] Chua F C, Ward J, Zhang Y, et al. Stringer: Balancing latency and resource usage in service function chain provisioning. IEEE Internet Computing, 2016, 20(6): 22-31.

[8] Saraswat S, Agarwal V, Gupta H P, et al. Challenges and solutions in software defined networking: A survey[J]. Journal of Network and Computer Applications, 2019, 141: 23-58.

[9] Bonfim M S, Dias K L, Fernandes S F L. Integrated NFV/SDN architectures: A systematic literature review. ACM Computing Surveys (CSUR), 2019, 51(6): 1-39.

[10] Amin R, Reisslein M, Shah N. Hybrid SDN networks: A survey of existing approaches. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3259-3306.

[11] Li C, Wu Y, Yuan X, et al. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. International Journal of Communication Systems, 2018, 31(5): e3497.

[12] Li Z, Xing W, Khamaisheh S, et al. Detecting saturation attacks based on self-similarity of OpenFlow traffic. IEEE Transactions on Network and Service Management, 2019, 17(1): 607-621.

[13] Zhuang W, Ye Q, Lyu F, et al. SDN/NFV-empowered future IoV with enhanced communication, computing, and caching. Proceedings of the IEEE, 2019, 108(2): 274-291.

[14] Yubo S, Xinyu Q, Qiang H, et al. Two-stage multi-classification algorithm for Internet of Things equipment identification. Journal of Tsinghua University (Science and Technology), 2020, 60(5): 365-370.