

A software defined security scheme based on SDN environment

Xiaolong Xu

School of Computer Science
Nanjing University of Posts and Telecommunications
Nanjing, China
xuxl@njupt.edu.cn

Liuyun Hu

State Key Laboratory of Information Security,
Chinese Academy of Sciences
Beijing, China
1216043127@njupt.edu.cn

Abstract—This paper analyzes the insufficiency of traditional network architecture in the current information era firstly, and introduces the concept of Software Defined Networking (SDN). Then it describes the problems existing in the traditional security protection by analyzing the importance of network security. On the basis of SDN, the paper analyzes the effects of the SDN technology on the traditional network security protection. Along with the idea of software defined, it puts forward the concept and the main idea of software defined security. By analyzing the classic architecture of software defined network, it is concluded that SDN technology was used to construct new network architecture to realize development and application of network virtualization. The current new network security architectures are analyzed, and it is concluded that the current security resolution schemes cannot adapt to the development of SDN. Therefore, it is necessary to build a new security architecture, which contains centralized management based on the SDN environment. The paper makes a detailed analysis of the architecture and the internal structure of security controller in the control layer. Then according to the security mechanism, the data flow process of the network security protection is described in detail. Finally, the paper analyzes the performance of the security mechanism in three security scenes and comes up with shortage of the mechanism.

Keywords-SDN;virtualization;Software Defined Security

I. INTRODUCTION

The hierarchical structure of the traditional network is the key to the great success of internet. However, with the expanding of network size, the traditional network device has too much complicated protocol, which increases the difficulty for operators to customize the network and holds back researchers to deploy new protocols in the real environment. At the same time, the rapid growth of the internet traffic and user's demands and the appearance of all kinds of new services, they increase the cost of network operating. Though current network architecture and network capacity are available, there is still a series of problems including stiffness in schedule and allocation of resource and slowness in the deployment of the new technology or application and high requirement of network operation and maintenance. With the development of information technology and network technology, the information system has become bigger and bigger and the

traditional network architecture has become very complex, which is difficult to manage. In order to meet the need of the high efficiency and flexibility, the innovation of traditional network architecture must come into effect, so Software Defined Networking (SDN) was born. In 2006, Stanford student Casado M and his mentor Professor McKeown N were inspired by the research project named Ethane [1], and then they presented the concept of OpenFlow. In 2008, McKeown N and others published a paper entitled: "OpenFlow: enabling innovation in campus networks" [2] in ACM SIGCOMM. In this paper, it firstly introduced the concept of OpenFlow, i.e., the separation of data plane and control plane in the traditional network devices, which controls management and configuration of all kinds of network devices by a centralized controller with a standardized interface. In 2009, McKeown N formally put forward the concept of SDN, whose core idea is keeping the control plane from network devices. By making use of the separation of control function and data plane, SDN realizes network programmability and network virtualization, and also forms a software programming defined network, which overturns current network communication and management pattern.

As the function of the computer system is increasingly perfect and the speed continuously improves, the system has become more and more complex, and its scale becomes bigger and bigger, which leads to the fact that access control and the number of logical connection is increasing. Network security is a prerequisite for the normal operation of enterprises and network. Network security is not only security for a single point but also security for the whole enterprise information network. Any security danger hidden in the current network system will cause the disruption of business network. So the importance of network security is obvious. At present, most of the enterprises and institutions have deployed firewalls and antivirus software, and some units also further have deployed VPN, intrusion detection system and the vulnerability scanning. To a certain extent, it reduces the threats of network security. In this paper, security devices consist of firewall, IDS, and so on. However, there are some problems existing in the traditional network security mechanisms:

- The deployment cost is too high. Traditional security is very dependent on the deployment of security devices, which is used to protect the

This work was jointly sponsored by the National Natural Science Foundation of China under Grant 61472192, the Talent Project in Six Fields of Jiangsu Province under Grant 2015-JNHB-012, the "333" Scientific Research program of Jiangsu Province under Grant BRA2017228, and the Scientific and Technological Support Project (Society) of Jiangsu Province under Grant BE2016776.

security of the information network. These devices are deployed in boundary of each branch network, which can only ensure the security of their own network, and it will cause that different network will deploy a large number of the same security devices, the redundancy of security devices is too high, which cannot deploy the security devices on-demand according to the network features.

- The linkage of security devices is not strong. In traditional network, each security device is independent, which only can prevent certain attack lonely by itself. So it cannot make deep protection effectively. Due to the types of security technology and manufacturers of security devices, devices are difficult to achieve the effective collaboration.
- The flexibility of security devices is not enough. The traditional network security technology focuses on the protection of the network and the reinforcement and protection of system. The main technical means solve the problem of network security by data identification technology, firewall technology, access control technology and so on. They all have a certain role in defending network intrusion, but the security technology of traditional network for network attacks mainly adopts the passive defense means, which has a great deal of hysteresis, it cannot timely and effectively protect network according to changes in network traffic, these technologies are not enough in the face of increasingly complex and ever-changing variety of intrusion.

In order to solve these problems, it is necessary to make use of SDN network's tractional data flow, which is flexible and efficient to call already deployed security devices to protect network information. There is a Software Defined Security (SDS) mechanism by using the idea of the separation of control and data. Since The famous consultancy Gartner has referred to the concept of Software Defined Security in "The Impact of Software - Defined Data Centers on information Security" [3]. The combination of Software definition and Security has become a security solution. Software defined security architecture introduces the thought of SDN architecture. It realizes the separation of the data plane and control plane, a software programming approach is useful for the application layer to intelligently, automatically and intensively manage the security resources, the control layer controls the underlying equipment and provides open interfaces to the upper through the controller, it invokes security devices according to security policy and the need, and then redesigns security devices in the data plane, which includes physical devices and virtual devices. It implement resource pooling of security devices by using the unified interface and tractional data flow in the SDN environment, and it realizes the combination and superposition of multiple security ability, which improves the overall protection efficiency of the system. The extensible and programmable SDS security framework

implements automation and coordination of security applications - security controller – security devices, which realizes a flexible and efficient security.

II. RELATED WORK

A. Description of SDN

Open Network Foundation (ONF) originally referred to SDN architecture [3] in the white paper, and ONF released the latest version in the SDN architecture overview version 1.0 [5] at the end of 2013. SDN architecture is shown in Fig. 1, which is divided into application layer, control layer and infrastructure layer (also called data layer). The infrastructure layer is mainly composed of OpenFlow switches which supports OpenFlow protocol and other network elements, the connections among various network elements is formed by SDN network data path of different rules. The control layer mainly contains a controller in logic center of the architecture, which is responsible for the implementation of the logical control strategy and maintains the view of entire network. SDN controller abstract the entire network view into web services, it is convenient for manager to customize personalized applications and to realize the logic management of network through the north API. The application layer contains all kinds of web application based SDN. Users do not need to care about the technical details of the underlying equipments, and they can achieve rapid deployment of new applications through a simple programming. The north interface API between application layer and control layer is responsible for communication. User can make the corresponding development in the application layer according to the different requirements, which provides more choices for manager. And infrastructure layer and control layer makes communication by using the control data plane interface. In order to promote the development of SDN, ONF released the unified standard of communication and mainly adopted OpenFlow protocol as the standardized application protocol at present. OpenFlow protocol is the basis to achieve the separation of control and forward, which describes the standards of communication information between controller and switches and the standards of interface between the control plane and data plane to realize the controller's centralized control of the whole network. The control layer is core of the whole architecture, and it has a session with underlying switches through OpenFlow protocol, and provides the interface for upper software in application layer, which is used to test status of network applications and to distribute control strategy.

SDN technology uses the thought of separation of data and control, which realizes the virtualization of network. It supports the wide application of cloud computing and virtualization technologies and brings disruptive changes to management of traditional network. Network topology, dynamic network and flow table can be programmed to make rapid changes to simplify network management, which brings great benefits to the network management.

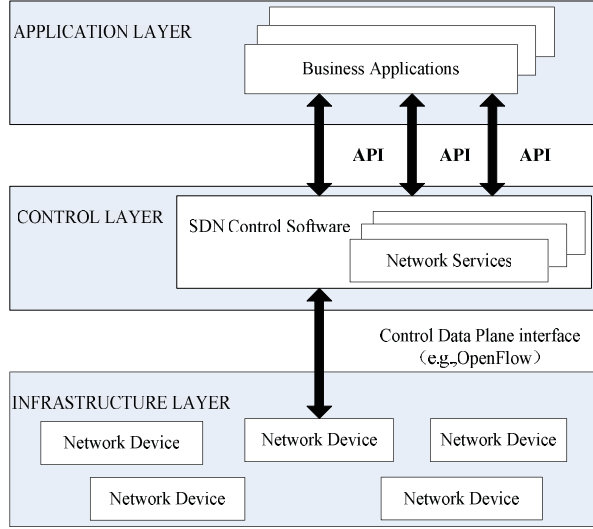


Figure 1. Software-Defined Networking Architecture

B. Current Security Architecture

From the perspective of security protection, the main network attacks can be divided into three categories: conventional network intrusion, denial of service and distributed denial of service (Dos and DDos), advanced persistent threat (APT) or directional attack [6].

These three kinds of typical network attacks that exist in the traditional network will also appear in the SDN network, which is also necessary for the SDN network security mechanism to face and solve. Traditional security protection is obviously unable to meet the demand of network virtualization security and to cope with the challenge of the above various security threats, due to the closed security architecture with single layer defense. It is necessary to need a new security solution, so the software defined security was born.

There have been some architectures concerning SDN currently, such as the FRESKO architecture [7], a security module joins in the SDN controller named NOX, which implements sixteen basic flow monitoring and threat detection function with python. It can use user script to detect SDN flow information, and quickly develops security products and services based SDN, which is open. But its application has no interaction with the underlying security devices, which cannot make deep protection. In addition, the security module integrates with SDN controller in the FRESKO architecture, and it is easy to make it real that excessive script becomes the bottleneck of SDN system.

Flow detection and analysis [8] is a viable method to detect attack in SDN environment, such as a detection method of distributed denial of service attack proposed by the literature [9], which obtains and analyzes the flow statistics regularly. Literature [10] proposes another flow analysis method, and it triggers a event named "flow was not found" when it checks flow table every time, and then

it send the encapsulating packet to SDN controller in form of the OpenFlow control messages named PACKET_IN, which checks whether the address of packet is mock, but this approach bring the controller huge load, and it is not extendable.

In addition, the testing of package is also very important in SDN security. References [11], [12] put forward how to analyze the application of flow. References [13], [14] detect attacks by using combination of the controller with the intrusion detection system (IDS). Reference [15] proposes an OpenFlow extension named FleXam, which samples for each flow, and it makes the SDN controller have access to part of the packet to determine whether there is attack, but this solution mixes data plane and control plane, it deviates from the original intention of SDN design. Qazi et al. proposes a reinforce strategy and protection plan [16], and it uses SDN application interface without modifying the existing devices, but SDN controller need to process some field of the flow to avoid conflict of address in multi-tenant environment.

In summary, the current SDN controller cannot cooperate with security devices, and depth detection of flow and packet cannot be achieved at low cost under the global view, which cannot realize the rapid and flexible response to the new security threats.

III. SOFTWARE DEFINED SECURITY SCHEME

A. Security Achitecture Based on the Security Controller

By adopting the idea of separation of control and data, there is a software defined security architecture in SDN network shown in Fig. 2 according to SDN framework and characteristics of the SDN network, which implements separation of the application, the control and the data, and looses coupling of devices. There are three layers: application layer, control layer, data layer (i.e., infrastructure layer). In the top of security architecture is the deployment of various APPs developed by third parties. In the middle of the architecture is control layer that includes SDN Controller and Security Controller (SC), the data layer at the bottom of architecture contains the network devices and security devices, which can be physical or virtual.

In the application layer of security architecture, there are all kinds of security APPs developed by third parties, which shows the security capabilities, it also centralizes security intelligence information. In this paper, security Apps consists of collection of security information, orchestration of security services, and so on. Security controller collects security intelligence information from the underlying devices, and then uploads to the application layer with generating alarm logs. The application layer can make new security protection according to this information. For example, by analyzing historical data, the abnormal traffic can be detected with alarms. Application plane has a very important purpose is the orchestration of security services, which automatically integrates each individual

security service. Combination and superposition of security services will be real in the architecture, which maximize the security ability of the existing services. Security service is called according to the arrangement service, which generates and pushes the fine-grained security strategy to control layer.

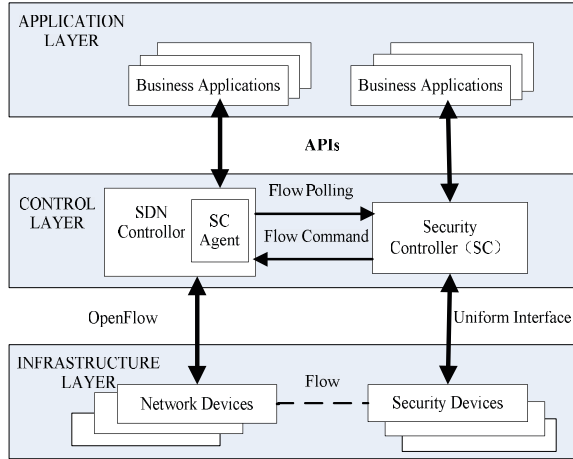


Figure 2. Software-Defined Security Architecture

In the control layer of security architecture, there is a dedicated security controller to complete security task. SDN controller is responsible for the scheduling of network traffic. In order to monitor packets in real time, it needs develop a security controller agent (SCAgent) in SDN controller, so security controller can interact with SDN controller through the west interface. For example, data flow, control commands and network topology information, and so on. The control layer establishes logging and credibility library with the security information from the data layer, at the same time it manages security devices (e.g., registration of devices). The devices cannot directly execute security policy distributed by the application layer, so the control layer need analyze security policy into specific commands for security devices. The control layer provides application layer and data layer with a unified interface, so as to receive data and schedule security resource. It generates security flow command and sends commands to SDN controller according to security policy, which realizes that data flow is redirected to the appropriate security devices.

In the data layer of security architecture, namely the infrastructure layer, there are some security devices, such as firewall, intrusion detection system, etc. Due to the resource pooling of security devices, its performance is a kind of security capabilities. All these devices will register their information to the security controller, for example, device ID, location and capacity, etc.

Security controller is the core of the control layer of security architecture, and its internal composition is shown in Fig. 3.

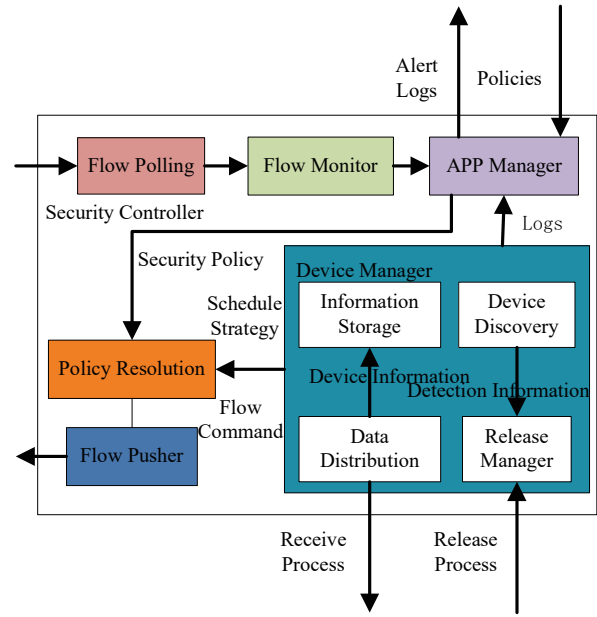


Figure 3. Security controller structure

Flow polling and flow monitor are the specific modules in the environment of SDN network. It has access to global flow information of current network by interacting with SC Agent. Then the flow monitor module detects suspicious behavior in the network, and sends it to the APP manager module in which flow monitor forms a warning log and posts to the APP of application layer.

APP manager receives security policy from the APP of application layer and orchestrates the requirements of APP security services, and it also informs the APP of warning logs. Policy resolution module translates the security strategy into flow command identified by SDN controller.

Flow pusher module, the flow command is pushed to SDN controller to issue the flow table to redirect data flow.

Device manager module, it stores the registration information of security devices and tests devices. It calls an appropriate security device from the all devices according to the scheduling strategy and maintains resource pooling of security devices. The device manager can be also divided into several parts, for example, information storage, device discovery, data distribution, release manager, etc.

In order to avoid that the load in some security devices is overweight, and that some of the security device is relatively free, it needs to develop appropriate scheduling policy, which will improve the utilization rate with reasonable distribution of the capacity of various security devices. There are several methods in SDN network: random algorithm, the fastest response, polling algorithm, weighted polling algorithm, least connections algorithm, and weighted least connection algorithm.

When security devices connect to the controller, it firstly registers to the device manager module, which records details of the device and stores the information in

the information storage module. Information storage module statically allocates the size of the memory to a security device for storing device information and data information. Although the allocation way has certain waste in the memory, it can improve the operation efficiency of the controller. At the same time, the device manager module creates two threads (releasing and receiving) for security devices, which is used to process the data of security devices.

When security controller receives the packet from security devices, the packet will be sent into the queue. The receiving thread is founded by device manager module when a security device is connecting to the controller, receiving thread reads packet from the queue and send it to the data distribution module.

Data distribution module parses the data packets. If it is information of the security devices, it will be sent to information storage module for processing. If it is the testing result of devices (when the attack is detected or protected by the security devices, warning logs will be pushed), it will be sent to the APP manager module to generate a warning log for uploading to the APP, which will build credibility library or take further security protection. If it is the renewal information of devices, it will be sent to the device discovery module to analyz, and device discovery process will be starting.

When the device discovery module receives packets or dateline of testing information is over, it will start the device discovery process. Firstly detection packet is generating and sent to the release manager module, which sends the packet to security devices to accomplish the device discovery process. When a device discovery process is over, the module will update device information in information storage module.

Release manager module package and process information that will be sent to the security devices, and it sends encapsulated packets to a queue of the security devices and send packets to security devices in turn.

B. Data Processing Cycle

Data flow is forward in accordance with the flow issued by the controller in SDN network, a flow table includes multiple flow tables, and each flow table item is as follows:

- Match field: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- Priority: matching precedence of the flow entry.
- Counters: updated when packets are matched.
- Instructions: to modify the action set or pipeline processing.
- Timeouts: maximum amount of time or idle time before flow is expired by the switch.
- Cookie: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion. Not used when processing packets.

Flow table item is decided by matching fields and priority, which decides the only flow table in a flow chart. Flow table is forwarding rules for data flow. OpenFlow assembly line processing defines the packet how to interact with the flow chart.

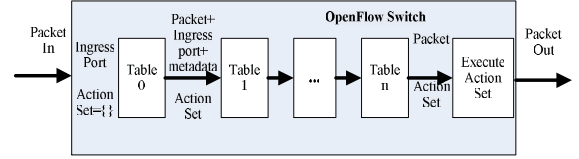


Figure 4. Packet flow through the processing pipeline

As shown in Fig. 4, the flow table in OpenFlow switch starts increasing number from 0. When packets are sent into the flow, it will firstly match the flow table item whose number is 0.

The instruction set in the flow table item is executed after matching, and then it executes a forward movement for data flow according to the instruction set.

Assume that the safety equipment in working condition had registered, all kinds of information of security devices has been stored in the list of the security controller.

When the terminal device needs to have access to the data center in the SDN network based on OpenFlow protocol, it will perform the specific process as follows:

Step1 : Terminal device wants to connect to a server of the data center, and it firstly sends the request information to the server's network system.

Step2 : After OpenFlow switch in the network system has received the request, it queries its flow table, and the matching process of data packets is shown in Fig. 4. If OpenFlow switch has a corresponding flow table, the request will be directly sent to the server without exception after security devices detecting. Terminal device can have access to resources after the server accepting requests. If there is no matching table item with the packet in the table, the switch will forward the packet to SDN controller by an event named Packet_In. When the switch sends a message to the controller, it can directly send the packet to the controller by the TCP protocol. It also can use encryption transmission for packets with transport layer security protocol (TLS) [17].

Step3 : SDN controller receives request information from OpenFlow switch, the information of flow is sent to security controller by the agent, flow monitor module tests data flow and sends it to the APP manager module for generating a warning log, security APP receives warning logs from the controller, and it will generate the corresponding security policy to issue to security controller.

Step4 : Security controller receives the security policy and analyzes it. The controller calls an appropriate security device and generates specific flow command according to

the registration information of safety equipment and device scheduling strategy.

Step5 : Flow command is sent to SDN controller by flow pusher module, then SDN controller issues appropriate flow table to the switch with an event named Packet_Out, which will set up routing and redirect the data flow to the appropriate security devices.

Step6 : The request of terminal device reaches security devices by routing after being detected. If the data flow is not unusual, the request of terminal device will arrive at the data center, and it has access to resources after data center accepting request. If the data flow contains exceptions, the results of being detected will generate warning logs by the controller and be uploaded to the APP that generates credibility library and takes further deep protection.

Step7 : When terminal device has access to the data center, there is the interaction of information, and this information will be sent to security devices for inspection, whose process likes as Step6.

Through the above steps, suspicious data is directionally tracted to the appropriate security devices for handling to protect the network security.

IV. ANALYSIS OF SOFTWARE DEFINED SECURITY SCHEME

In this section, data center based on SDN network and the deployment of security devices are described. SDN technique has just emerged in recent years, which leads to the result that SDN devices are too expensive. In addition, the virtualization of security devices is widely used. So the test can be carried out in virtual environment. Mininet [17] is a good simulation software that can quickly create a complete set of SDN networks, and all of its commands can be applied to the real SDN networks. Therefore, data center based on SDN network can be simulated in Mininet, and then it connects with the remote SDN controller.

In order to build the data center based on SDN network, the necessary softwares must be installed first, the operating system of virtual machine and host is ubuntu14.04LTS. Mininet is installed in the virtual machine, and the controller named OpenDayLight [17] is installed in the host. Mininet has many network topologies which consist of tree, single, linear and minimal. By default, a minimal topology is created, which just has a switch and two hosts. In addition, the user can redesign the network topology by using python, which will be invoked by the python API of Mininet.

Fat-tree [17] is used to realize the data center in this paper. Fat-tree is an improvement based on the traditional tree structure, in which switches are in center. There are three kinds of switches in Fat-tree, which consists of core, region and edge. An intermediate node can have more than one parent node, which increases the number of links between switches. Due to increasing the connectivity of network, Fat-tree improves the reliability of the data center. A k-fork Fat-tree is build, and the region layer and the

edge layer are divided into k pods. Each pod is a structure of two-layer switches, which consists of $k/2$ region switches and $k/2$ edge switches. In this section, a Fat-tree network topology with 2 pods is designed, whose core code is as follows:

```
# add core ovs
for i in range( L1 ):
    sw = self.addSwitch( 'c{}'.format( i + 1 ) )
    c.append( sw )

# add aggregation ovs
for i in range( L2 ):
    sw = self.addSwitch( 'a{}'.format( L1 + i + 1 ) )
    a.append( sw )

# add edge ovs
for i in range( L3 ):
    sw = self.addSwitch( 'e{}'.format( L1 + L2 + i + 1 ) )
    e.append( sw )

# add links between core and aggregation ovs
for i in range( L1 ):
    sw1 = c[i]
    for sw2 in a[i/2::L1/2]:
        self.addLink( sw2, sw1 )

# add links between aggregation and edge ovs
for i in range( 0, L2, 2 ):
    for sw1 in a[i:i+2]:
        for sw2 in e[i:i+2]:
            self.addLink( sw2, sw1 )

#add hosts and its links with edge ovs
count = 1
for sw1 in e:
    for i in range(2):
        host = self.addHost( 'h{}'.format( count ) )
        self.addLink( sw1, host )
        count += 1
```

The topology created in Mininet is shown in Fig. 5

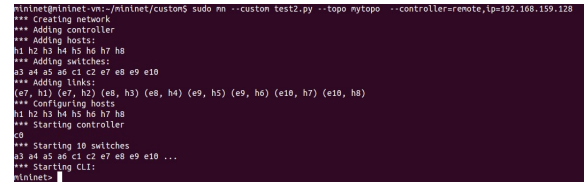


Figure 5. The topology created in Mininet

Softwares related to security devices are installed in the virtual machine and they connect to the data center by bypass deployment. There is no uniform standard for the interface between SDN controllers. In order to realize the interaction between the controllers, the communication between the controllers can be realized by using cluster technique (such as Hazelcast).

With the development of the virtualization technology of security devices, various attacks can be tested in this test bed in the future. The research of security controller and the pool of security have been in the initial stage. In addition, the uniform management of security devices has not been realized. So the process analysis of various attack scenarios is conducted in this section.

Considering the above situations, the test bed is shown in Fig. 6.

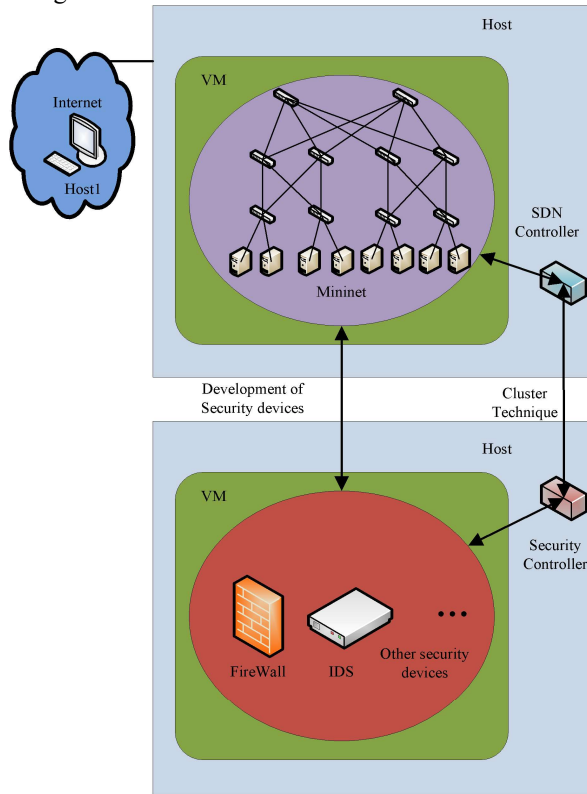


Figure 6. Test bed

In the test bed, OpenFlow switches are as network access points for all kinds of network. The terminal devices have a communication by SDN network, and SDN virtualization technology can ensure the security of data transmission. OpenFlow switches connects SDN controller through a secure channel, they register their information to the controller, so the controller will master the network topology and the information of global flow. Network security device is deployed in the boundary of each branch network (e.g., firewalls), but it can only protect the security of its network. In this scheme, security devices connect to the controller through a unified interface, and they register information to the security controller (such as

device ID, location of device, device type, etc.), which forms resource pooling of security services. A large number of same security devices do not need to be deployed at the import and export of each access network, which saves the cost of the deployment of security devices, resource pooling only performs security capabilities through the security controller. SDN controller flexibly controls the direction of data flow, and the appropriate flow table is issued to the OpenFlow switch, so the suspicious data flow is redirected to the appropriate security devices. Security resource pooling flexibly protects each network with cooperation of the two above.

There is a simulation network intrusion scene in the Fig. 6. Assume that security devices (all common security devices) have registered themselves information to the security controller.

- Scene 1 Regular Network Intrusion

Step1 : Host 1 wants to have access to the data center, and it firstly sends the request information to the network system.

Step2 : Network system receives request information from Host 1. OpenFlow switch queries whether there is a corresponding flow table. If there is a matching flow table item, the request will be directly sent to intrusion detection system (IDS) for detecting, the request information without exception will arrive at data center for access to resources. If there is no corresponding flow table item in OpenFlow switch flow table, the request will be forwarded to SDN controller.

Step3 : SDN controller receives the request, and it interacts with security controller through SCAgent. Flow polling module obtains the corresponding flow, and flow monitor analyzes data and informs the security APP. Security APP receives warning logs from the controller and it will generate the corresponding security policy, which is issued to security controller.

Step4 : Security controller receives the security policy for strategy analysis, and queries registered security devices. According to the deployment of resource pooling of IDS, it uses appropriate scheduling algorithm and calls the IDS device that is in working condition and of less load for generating specific flow commands.

Step5 : Specific flow command is pushed to SDN controller and instructs SDN controller to issue flow table to switch for establishing proper routing.

Step6 : The request of Host 1 reaches IDS device by routing. After being detected in IDS device, if the data flow is normal, the request of Host 1 reaches data center, Host 1 has access to data center for resources after the server accepting requests. If data flow contains exceptions, IDS will use detecting results for generating warning logs and post to security controller, and then security controller uploads it to the APP, the APP issues security policy to controller. According to the deployment of security devices, the policy instructs SDN controller to generate

flow table to remove abnormal flow table or take deep protection.

Step7 : In the process of Host 1 having access to the data center, interactive information will be sent IDS for detection, the processing methods likes Step6.

- Scene 2 Distributed Denial of Service

Step1 : Flow polling module of security controller obtains flow information of a server through SCAgent.

Step2 : When flow monitor module has detected DDOS attack, it generates the warning logs in time and uploads security APP in application layer by the APP manager module.

Step3 : APP receives suspicious data from security controller and adopts corresponding measures for issuing the corresponding security policy to security controller.

Step4 : Security controller will parse security policy and query resource pooling of security devices. According to the scheduling algorithm, it calls suitable DDOS cleaning system, and then generates specific instructions and sends it to SDN controller.

Step5 : SDN controller issues proper flow table to the switch according to the receiving instructions and establishes proper routing.

Step6 : After routing being established, the controller will schedule the flow of the server to flow cleaning system for cleaning, and the traffic reinjects after cleaning.

Step7 : After the above steps, the server will be able to receive the normal request.

- Scene 3 Advanced Persistent Threat

The data center is a protection goal with the high value and security level of Host 1 is low, and it needs to have access to the data or services of center data.

Step1-Step6: Similar to the Step1 - Step6 in Scene 1.

Step7 : In the process of Host 1 having access to the data center, there is interactive information. According to security policy, security controller instructs SDN controller to schedule the data of Host 1 to Deep Packet Inspection (DPI) device and applications identify services. At the same time, the flow monitor module analyzes the network behavior and timely uploads warning logs, once the Host 1 is identified as a suspicious host, its flow will be sent to the sandbox for segregation or to the honey pot in accordance with the security policy.

V. CONCLUSION

This paper expounds the architecture and environment characteristics of SDN and puts forward a hierarchical centralized software defined security scheme based on SDN environment, which implement the pooling of the security resources. It combines SDN controller with the coordination work of security APP, security controller and security devices implement the directional traction of data flow to security devices, which detects all kinds of security threats of SDN environment, and then according to the

security policy, it makes deep analysis and rapid response to protect the network security.

However, the technology of SDN is not enough mature and software defined security is still in research stage, the specific implementation detail of security controller in this paper has not yet be completed. The interaction has not formed standard between the controllers. In the future research, there will be breakthrough of the above two, with the development of SDN technology and the further research of SDS, this mechanism will be applied to the new network.

ACKNOWLEDGMENT

We would like to thank the reviewers for helping us improve the quality of this paper. This work was jointly sponsored by the National Natural Science Foundation of China under Grant 61472192, the Talent Project in Six Fields of Jiangsu Province under Grant 2015-JNHB-012, the "333" Scientific Research program of Jiangsu Province under Grant BRA2017228, and the Scientific and Technological Support Project (Society) of Jiangsu Province under Grant BE2016776.

REFERENCES

- [1] M. Casado, M. J. Freedman, J. Pettit, et al., "Ethere: taking control of the enterprise," *ACM SIGCOMM Computer Communication Review*, vol. 4, no. 37, pp. 1-12, 2007.
- [2] N. McKeown, T. Anderson, H. Balakrishnan, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 2, no. 38, pp. 69-74, 2008.
- [3] N. MacDonald, "The Impact of Software-Defined Data Centers on Information Security," Gartner Inc, USA, Rep. G00229660, 2012.
- [4] Software- Defined networking: The new norm for networks, Open Networking Foundation, 2012.
- [5] SDN architecture overview, 1st ed., Open Networking Foundation, 2013.
- [6] L. Pingree and N. MacDonald, "Best practices for mitigating advanced persistent threats," Gartner Inc, USA, Rep. G00224682, 2012.
- [7] S. Shin, P. Porras, V. Yegneswaran, M. Fong, et al., "FRESKO: modular composable security services for software-defined networks," in *Proc. 2013 Network and Distributed Security Symposium*, San Diego, USA, 2013, pp. 135-139.
- [8] A. Hassidim, D. Raz, M. Segalov, et al., "Network utilization: the flow view," in *Proc. 2013 IEEE Int. Conf. on Computer Communications*, Turin, Italy, 2013, pp. 1429-1437.
- [9] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. 2010 IEEE 35th Conf. on Local Computer Networks*, Denver, USA, 2010, pp. 408-415.
- [10] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in *Proc. 19th IEEE Int. Conf. on Network Protocols*, Vancouver, Canada, 2011, pp. 7-12.
- [11] Z. A. Qazi, J. Lee, T. Jin, et al., "Application-awareness in SDN," in *Proc. 2013 ACM SIGCOMM Conf. on SIGCOMM*, Hong Kong, China, 2013, pp. 487-488.
- [12] M. Jarschel, F. Wamser, T. Hohn, et al., "SDN-based applicationaware networking on the example of YouTube video streaming," in *Proc. 2nd European Workshop on Software Defined Networks*, Berlin, Germany, 2013, pp. 87-92.

- [13] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Int. Workshop on Recent Advances in Intrusion Detection*, Berlin, Germany, 2011, pp. 161-180.
- [14] S. Kumar, T.Kumar, G.Singh, et al., "Open flow switch with intrusion detection system," *Int. J. Sci. Res. Eng. & Tech.*, vol. 7, no.1, pp. 1-4, 2012.
- [15] S. Shirali-Shahreza and Y. Ganjali, "Empowering software defined network controller with packet- level information," in *Proc. 2013 IEEE Int. Conf. on Communications Workshops*, Atlanta, USA, 2013, pp. 1335-1339.
- [16] Z. A. Qazi, C. C. Tu, L. Chiang, et al., "SIMPLE-fying middlebox policy enforcement using SDN," in *Proc. 2013 ACM SIGCOMM Conf. on SIGCOMM*, Hong Kong, China, 2013, pp. 27-38
- [17] T. Dierks and E. Rescorla. (2008,Aug). *The Transport Layer Security (TLS) Protocol(Ver. 1.2)*[Online]. Available: <http://tools.ietf.org/html/rfc5246>
- [18] Mininet. [Online] Available: <http://mininet.org/>
- [19] OpenDayLight SDN controller. [Online]. Available: <https://www.opendaylight.org/>
- [20] C. L. Leiserson "Fat trees: Universal networks for networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. C-34, no. 10, pp.892-901, 1985