Defending Blind DDoS Attack on SDN Based on Moving Target Defense

Duohe Ma[1,2(✉)], Zhen Xu[1], and Dongdai Lin[1]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
{maduohe,xuzhen,ddlin}@iie.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Software Defined Networking (SDN) provides a new network solution by decoupling control plane and data plane from the closed and proprietary implementations of traditional network devices. With its promisingly advanced architecture, SDN represents the future development trend of network. In its typical structure, collaborative interaction between one controller and multiple switches forms a centralized network topology. As playing a key role in this network architecture, the controller in SDN is very vulnerable to single point of failure. What is worse, the emergence of Blind DDoS attack against SDN's special structure increases its risks. To address this challenge, we introduce a Moving Target Defense(MTD) system to defend Blind DDoS attack. The approach adopts a multi-controller pool to solve the saturation problem, and it can dynamically shift controllers connecting to switches according to the density of flood flow. By randomly delaying the scanning packets and filtering the flood with route-map, this MTD system can effectively resist the Blind DDoS attack and protect the availability and reliability of SDN.

Keywords: Blind DDoS attack · Software defined networking · Moving target defense

1 Introduction

The core idea of software defined networking (SDN) [1] is to abstract and decouple control plane and data forwarding plane, making network management and expansion more flexible [6,8]. The structure of SDN is divided into centralized controller and forwarding device (e.g. switch). The controller is responsible for management, control and configuration of network devices using standard protocol such as OpenFlow [2,3]. It also issues flow rules generated thereof to switches through secure channel. Switches maintain flow table and forward network data according to flow rules. Switches receive querying instructions sent by the controller to report the network state. The OpenFlow technology is currently one of successful implementation under the SDN conception. In addition, Protocol

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 463–480, 2015.
DOI: 10.1007/978-3-319-23829-6_32

Oblivious Forwarding (POF) [25] architecture put forward by Huawei is also a material implementation of SDN idea.

In whatever implementation of SDN, the controller always plays the core function in SDN system and is the most vulnerable part and the weakest link in the whole SDN system security chains. Single point of failure is a very common security threat to centralized model controller [7]. It may be induced by a number of factors including physical damage, communication line failure, and a variety of attacks. SDN controller is an assembly of control surfaces. There are many instructions between the controller and switches. In case the switch receives initial packets, it will forward these packets to the controller. In a complex network environment, either bandwidth resource or computing resource of the controller may turn out to be bottleneck of the SDN system. Especially in OF_ONLY mode, switches are heavily dependent on the controller, so the entire network will be paralyzed when the controller is in breakdown.

Besides the above-mentioned shortcomings, SDN controller is also vulnerable to DDoS attacks. Traditionally, an attacker may directly launch DDoS attack on any network host on condition that the attacker has detected its IP address [26]. When it comes to SDN, there is an extra way to launch DDoS attack. The attacker sends a large number of packets to the switch that cannot be processed, which all will be forwarded to the controller by the switch according to OpenFlow protocol. When packets from multiple switches flood to the controller, the controller's processing competence will degrade. More seriously, denial of service will occur as a result. For this kind of attack, the attacker needs not know the IP address of the controller. In other words, the attacker can launch DDoS blindly. Thus it is a specific new DDoS attack on SDN architecture and we call it *Blind DDoS*. As composed a closed system with the controller and switches, SDN can avoid *Direct DDoS* attack by hiding information of its topology. This paper focuses on Blind DDoS attack and its defense.

In order to solve the above mentioned problem, this paper proposes a multiple controllers security method based on Moving Target Defense (MTD), which adopts a strategy to run a number of dynamically extensible controllers in SDN architecture. Even in the scanning stage, the packets' response time will also be changed dynamically by MTD strategy. The remaining sections of this paper is organized as the followings: Sect. 2 is an analysis of the principles of Blind DDoS attack including its generation process, harms and characteristics; Sect. 3 presents a novel MTD model as well as a multi-controller MTD system based on this model. In Sect. 4, the MTD defense approach is tested and evaluated. In Sect. 5, we will talk about the limitations of our approach and give the recommendations to improve them in the future works. Section 6 provides a comparative analysis between this paper and related researches. In the last section, a summary of this paper is presented.

2 Blind DDoS Attack

Taking OpenFlow for example, SDN switch forwards packets in accordance with flow table rules, where the fresh packet or abnormal packet that cannot be

processed in the flow table will be sent to the controller. In this sense, there is no need for an attacker to catch the IP address or location of the controller through scanning before launching an attack. Since as long as the attacker sends some specific attack packets and abnormal packets to SDN networks, all switches will automatically forward these packets to their controller.

Comparing with traditional DDoS attacks [5] which need to exploit victim host's IP addresses at first, this kind of DDoS on SDN controller is blind. So we define it as Blind DDoS attack. Paralysis of the controller as a result of data flow eruption sent to SDN network marks successful implementation of a Blind DDoS attack. Figure 1 gives the general view of Blind DDoS attack.

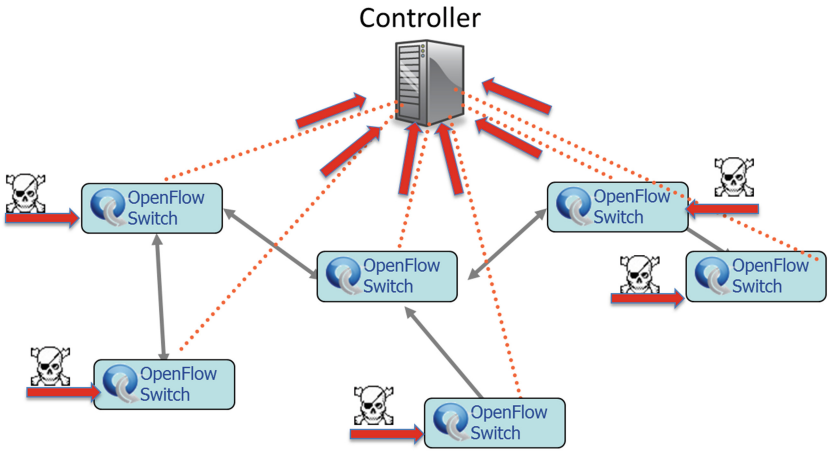

Fig. 1. Blind DDoS attack on SDN controller

Every flow entry in the flow table of a switch contains three items, i.e. *rule*, *action* and *stats*. The attacker can make a new or abnormal packet from carefully selected IP, Port, MAC etc. and then send it to the switch. Generally, there is no rule in the switch matching the fresh packet sent for the first time. The packet will be uploaded to the controller, and then controller will broadcast this packet's information to all network interfaces to find it's route. Once getting the route, the controller will issue corresponding rules to the switch' flow table. Otherwise, the controller will make a rule to switches to drop these packets. This whole response process will take a long time. Then the attacker will send a group of packets with the same information for a second time to the switch, if the response time is much shorter than that of the first time, the network can be determined to be SDN architecture. An attacker may launch Blind DDoS directly on the network which claims to be SDN network architecture or which the attacker has already known is SDN system by scanning (Fig. 1).

Blind DDoS attack is a serious threat to the security of SDN. On the one hand, a great quantity of attack data flow may cause the flow table of the switch to be full of rubbish rules, resulting in performance degradation or flow table entries overflow. On the other hand, Blind DDoS attack will cause network

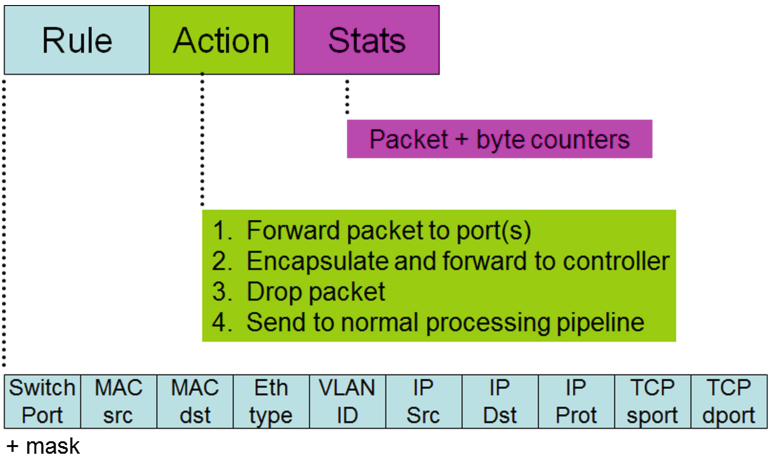

Fig. 2. Flow entry structure in SDN switch [2]

paralysis by causing the controller work improperly. Traditional network security methods provide no effective defense against this kind of attacks. Therefore, it needs development of new defense method to reduce its threat to SDN.

3 Moving Target Defense Method Against Blind DDoS

Existing defense systems including Firewall, IDS, IPS, WAF etc. all adopt static passive defense technologies, as a result, they are unable to provide dynamic security defense effectively against unknown or instantaneous attacks on the network. Most defense systems are devoted to pursue perfect detection and to prevent all attacks. However, it is clearly not rational because there are endless zero-day vulnerability like *Openssl' Heartbleed* on April 9, 2014. Therefore, network security researchers are actively exploring new security model [17–19], in pursuit of steady balance between security and defense costs. Moving target defense is one of these achievements which is completely different from previous Detection-based network security model.

3.1 Concept of MTD

As a fresh kind of defense, moving target defense does not seek to establish a perfect system to fight against all attacks. In practice, the idea of moving target defense is constantly diversing or changing the target to reduce the chances of vulnerability exposure, which will increase the attack difficulty and costs of the attacker. In essence, moving target defense technology realizes protection of objects by moving them.

In information attack and defense scenarios, the moving target defense system consists of method, channel, data and other resources. In [24], attack surface

is defined by means of formal description and used as the main reference for modeling of moving target defense. Attack surface is made up of method, channel, data and other resources that may be exploited by the attackers. Attack surface's features include IP address, ports, identity of the host, program language and data, etc. A moving target defense can be modeled using an attack surface together with different shifting strategies. Moving target defense may be divided into defenses at network layer [20], application layer, software layer [21], system layer and other layers corresponding to layers of the attack surface features. When automatically shifting the system's attack surface by changing one or more features, the target becomes unpredictable for the attackers. Constant changing attributes will increase attack difficulty and costs for the attackers. It will effectively reduce the chances of vulnerability exposure as well as the chances of being attacked and increase flexibility of the system.

However, attack surface only describes static properties of the target system, while fails to define or describe how the attack surface shift, the space of each property to shift or the shift frequency. It neither takes the overall characteristics of target system nor confederates the attackers. Thus, current MTD model based on attack surface is far from perfect.

3.2 A Novel MTD Model

Mandhata et al. [24] proposed a concept of system attack surface and gave its formal definition as the followings.

Definition 1. *The environment of system s,* $E_s =< U, D, T >$*, wherein U is the user, D is data storage and T is systems other than s in the set of global system S, i.e.,* $T = S/\{s\}$.

Definition 2. *As a specified system s and its environment* E_s*, the system's attack surface of s includes* $< M^{E_s}, C^{E_s}, I^{E_s} >$*, wherein* M^{E_s} *is a set of inlets and outlets of the system s,* C^{E_s} *is a set of channels of system s and* I^{E_s} *is a set of untrusted data entry of system s.*

According to the definition of system attack surface, reduction in the number of features of attack surfaces can enhance the security of system s. In a MTD system based on attack surface in the premise of keeping system service unchanged, the number of features is not reduced, rather, attack surface is shifted. Elements in every feature set in system attack surface are replaced so as to increase the difficulty for the attackers to guess the properties of these elements being used, consequently making it difficult for them to implement attacks.

In essence, moving target defense makes it difficult for the attackers to launch attacks exploiting the attack surfaces by means of constantly changing them. Therefore, randomization of the features' elements or attack surface shifting strategy is the key point of moving target defense model building. Hoverer, the MTD model built by shifting attack surfaces of the system has many defects, including mainly the following aspects:

(A). Although system attack surfaces have defined three sets, i.e. M, C, I, etc., and each set contains a plurality of elements, alternative variables for each element are not given, namely the shifting space for elements are not defined.

(B). The shift frequencies for each set or element are not specified for attack surface shifting strategy.

(C). The type of system s is not considered, though s may be a fully open system (such as web service), fully closed system (such as hosts in IPSec VPN) or a semi-open and semi-closed hybrid system.

(D). The attacker's actions and policies are not considered.

To solve the above problems, this paper presents a novel MTD model, which is the basis for design of SDN defense system against Blind DDoS attack proposed in the following parts of this paper.

Definition 3. *We propose a novel MTD model which has 3 tuple,*
The New MTD: $< S^{<N,R,T>}, A^{<G^a>}, D^{<G^d,F^d>} >$*, wherein,*

$S^{<N,R,T>}$ *is a target system,* $A^{<G^a>}$ *is an attacker,* $D^{<G^d,F^d>}$ *is a defender;*

$N = \{n_1, n_2, ..., n_i\}$*is the attack surface of system* S*,while* n_i *is the elements of attack surface;*

$R = \{r(n_1), r(n_2), ..., r(n_i)\}$ *is shift space for the elements* n_i*;*

$T = \{O, C, H\}$ *is three types of a system, where* O *represents full open system,* C *represents fully closed system and* H *represents semi-open and semi-closed hybrid system.* $l\, G^a = \{g_a(1), g_a(2), ..., g_a(i)\}$*is a set of attack strategies of* A*;*

$G^d = \{g_d(1), g_d(2), ..., g_d(i)\}$*is a set of defend strategies of* D*;*

$F^d = \{f_d(1), f_d(2), ..., f_d(i) | f_d(i) \rightarrow g_d(i)\}$*is the shift frequency of every strategy;*

Below is a case study of MTD Model, taking defense against Blind DDoS attack on SDN for an example.

SDN is a semi-open and semi-closed hybrid system, where the switch is open to the attacker and the user, and the controller is closed and invisible to the attacker and the network user.

For a closed system, legitimate users may access it by authorization and authentication, thus shift frequency for features of MTD model in it are not

Table 1. MTD Model in SDN

Feature	Values
N ·	$\{n_1$=direct I/O with switch, n_1 = indirect I/O with controller$\}$
R ·	$\{r(n_1)$=packets received by switch, $r(n_2)$ = available IP of controller$\}$
T ·	$\{H \mid (Attacker - Switch) \rightarrow Open, (Attacker - Controller) \rightarrow Closed\}$
G^a ·	$\{g_a(1)$=SDN-Scan, $g_a(2)$ = Blind-Flood$\}$
G^d ·	$\{g_d(1)$=randomly delay packets, $g_d(2)$ = randomly select controller$\}$
F^d ·	$\{f_d(1), f_d(2) \mid f_d(1) \rightarrow g_d(1), f_d(2) \rightarrow g_d(2)\}$

required to be too high, rather, it is proper as long as it can prevent force attacks. In an open system, as a large number of legitimate users and attackers mixed together are hard to be distinguished or granted authorization, and there is a possibility of distributed force guess in a short period, the elements of the attack surface has to shift every interaction. In a semi-open and semi-closed hybrid system as SDN, the two principles of closed system and open system mentioned above should be applied together. In a SDN system, the switch is open for an attacker while the controller is closed. When the attacker tries to launch a scanning attack based on response time difference, the switch may randomly delay the transmission and feedback time of the packets which match the flow table rules to achieve MTD, confusing information received by the attacker. Packet-delaying operation requires applying on each packet (e.g. $f_d(1)$ in Table 1). For the purpose of defending Blind DDoS attacks, as the controllers are closed, their shift frequencies (e.g. $f_d(2)$ in Table 1) are not required to be too high, whereas the shift space shall be large enough to prevent statistical attacks.

3.3 Implementation

MTD system proposed herein comprises the following components: a controller-pool consists of a number of controllers, MTD strategy manager, Flood-Filtering equipment based on route-map rules and SDN switches. Its architecture figure is as Fig. 3.

The controller-pool maintains multiple controllers, which can be physical machines or virtual machines. One controller which working as online is set to *master model* and all other controllers which working as offline are set to *equal model*. Generally, only one controller is online while other controllers are offline. In case the controller online has detected the number of packets which can not be routed beyond the default threshold, it will notify MTD strategy manager to start a number of controllers from offline to online.

MTD strategy manager is responsible for monitoring online controller's bandwidth and load. When alarming of the controller is trigged by Blind DDoS attack data flow, MTD strategy manager will shift multiple offline controllers to online status and assign appropriate IP addresses to them. And MTD strategy can change the controller's role between *master* and *equal* by sending $Role-Change$ messages to the switch. The controller initially online will issue to the switches a series of configuration instructions for defense of attacks. When Blind DDoS attacks stop, the number of online controllers should be drop.

MTD strategy manager will send two instruction to switches when there is Blind DDoS attack. One defense configuration instruction received by the switches is setting last rule in the flow table as default so as to forward all packets which do not match flow table rules to Flood-Filtering equipment rather than report them to the controller. We adopt *Bloom Filter* [28] method in Flood-Filtering equipment to improve the matching speed. The other defense configuration instruction is to randomly select a new controller for communication by sending *Role-Change* messages to the switch.

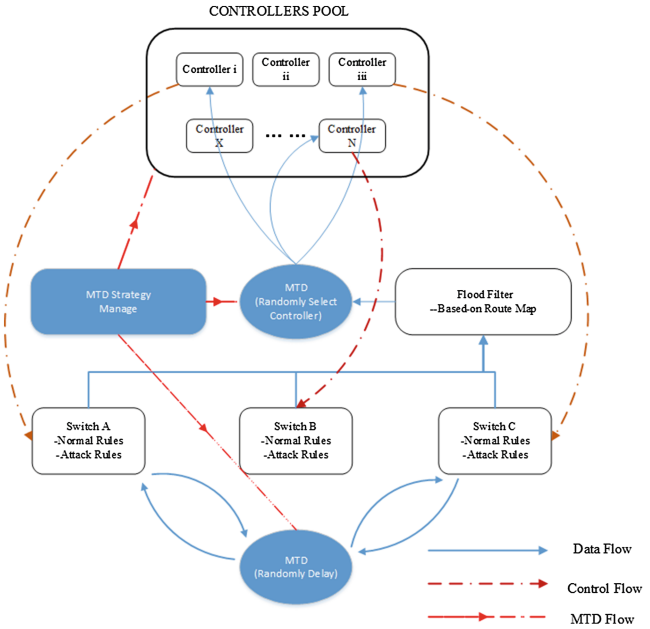

Fig. 3. The architecture of MTD

In addition to filtering common protocol vulnerability attacks, Flood-Filtering equipment also maintains all network's routing information and verify the validity of packets' destination IP to filter malicious forged packets of Blind DDoS attack. MTD strategy will continue to update the route-map rules from the controllers online and the route-map rules will be maintained for a long time.

4 Experiment and Evaluation

In the experiment, we adopt OpenvSwitch serveing as the switch, Floodlight [14] as SDN controller and PC with route-map matching software as Flood-Filtering device, all of which installed on X86 Pc with Intel(R) Core(TM)2 Duo CPU 2.40 Ghz, 2 GB RAM memory and CentOS 6.3. A windows server2003 with Apache Tomcat is used as a web service. IXIA equipment is used for generation of attack data flow and background flow. MTD manager is applied on controllers (Fig. 4).

Blind DDoS attack simulation is divided into two stages, where at the first stage, the attacker launches scanning attacks on network to confirm whether it is a SDN and at the second stage, the attacker sends flood packets of Blind DDoS to a SDN system.

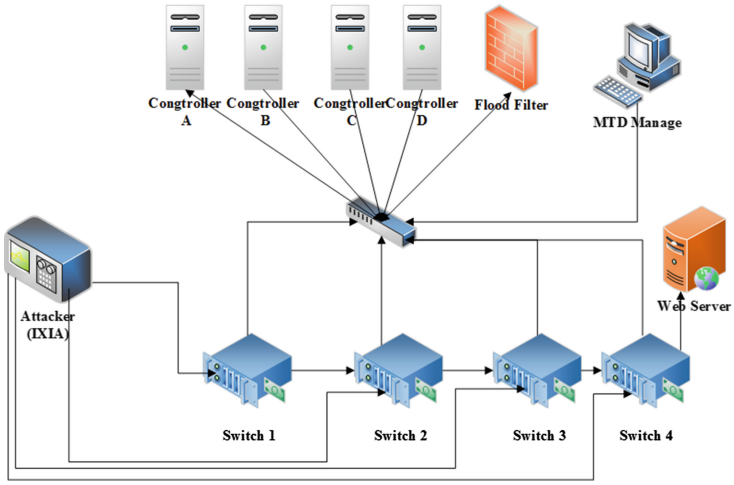

Fig. 4. Examination topology

4.1 Attack Stage I

Here we define *FirstPacket* and *LastPacket* which will be used in follow sections. In SDN, first several packets can not be routed by swithes because there are no rules to match these fresh packets. So the response time will be longer than the following packets. In our experiment, the number of these *ping* packets ranges from 1 to 9, with median 5. We use *FirstPacket* to stand for one of these initial packets and *LastPacket* to stand for one of following packets. The response time of *FirstPacket* is t_1 and that of *LastPacket* is t_2.

At the stage of scanning attack, whether it is a SDN network is mainly judged by the time difference between the network's response times to the packet sent for the first time t_1 and the same kind packet sent for the second time t_2. In traditional Network, t_1 is nearly equals to t_2 as showing in Table 2. But in

Table 2. Scan packets response time in traditional network (ms)

No	1	2	3	4	5	6	7	8	9	10	11	12
FirstPacket	0.989	0.975	1.04	1.054	0.868	0.861	1.017	1.019	1.06	1.07	1.023	0.908
LastPacket	0.982	1.025	0.654	1.08	0.703	1.281	0.804	0.948	0.953	1.019	0.671	1.018

Table 3. Scan packets response time in SDN (ms)

No	1	2	3	4	5	6	7	8	9	10	11	12
FirstPacket	4.46	5.67	5.86	4.05	4.05	3.57	4.05	3.52	4.17	4.08	3.9	3.77
LastPacket	1.069	1.17	1.015	0.771	1.04	0.959	0.804	0.984	0.957	1.083	0.995	1.02

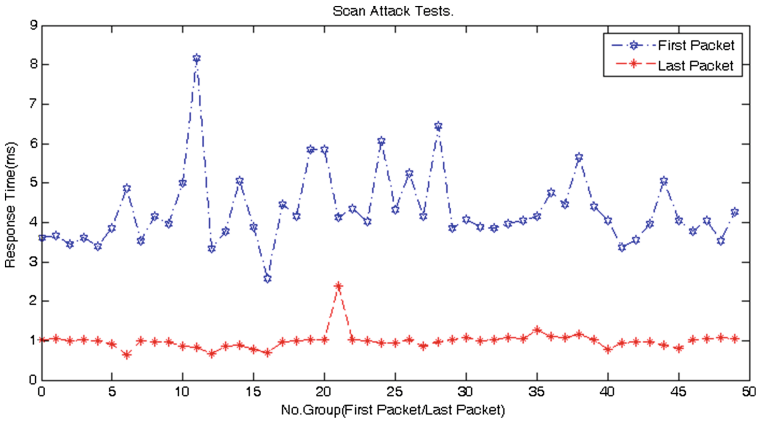

Fig. 5. Scan attacks

SDN, there are huge differences of the response time between $FirstPacket$ and $LastPacket$ as showing in Table 3.

Figure 5 shows the Scan Attacks result with slow rate of *ping* to the host of web service. For the purpose of combating scanning attack, MTD manager will make up a MTD *Random Delay* strategy (strategy $g_d(1)$ in Table 1.) according to the test results, the controller will deliver that strategy to the switch for the latter to randomly prolongs t_2 for a period time when processing packets matched with the flow table rules, so that $(t_1 - t_2)$ will approach 0.

We define D_t as the response time difference of $FirstPacket$ and $LastPacket$:

$$D_t = \{d_{t(i)} | d_{t(i)} = t_1(i) - t_2(i), i > 0\} \tag{1}$$

It can be easily proved that D_t has relation with both the SDN structure and the enter point, regardless of the client. So we give the MTD strategy of d_1 in MTD model showing in Table 1 with randomly delay packet as T_2:

$$T_2 = \left\{t'_{2(i)} | t'_{2(i)} = t_2(i) + Random[Min(D_t), Max(D_t)], i > 0\right\} \tag{2}$$

Figure 6 shows that the response times of packets were confused by the switch with MTD randomly delay strategy. So it will be hard to make a difference between SDN response time and traditional network response time.

4.2 Attack Stage II

In our simulation experiment, DDoS attack flow is generated by IXIA. Provided that attack flow stays unchanged, the effect of DDoS attacks is correlated with the following two factors, i.e. size of the data packet and randomness of the destination IP. As shown in the first figure, the effects on performance of target host's CPU by attacks through TCP Flood, UDP Flood, ICMP Flood and

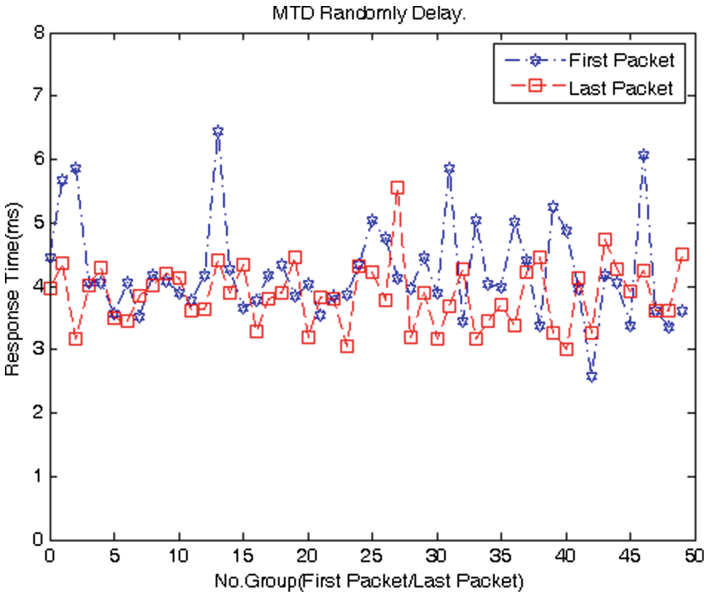

Fig. 6. Packets with MTD randomly delay

Flood without protocol in the same flow size and packet length are just slightly different. For the same kind of protocol, under constant attack flow, experiments with data packets in 64 Bytes and 1024 Bytes at the same rate 800 Mbps show that data packets in smaller size are more hazardous to target host than those in bigger size (Fig. 7).

If destination IP address of the attack data packet is matched with rules in the flow entry of the switch, the attack flow will not be sent to the controller; consequently, Blind DDoS attack will be ineffective. The following figure shows the data packets received by the controller in conditions of Destination IP and Random Destination IP DDoS attacks with packet size 1024 Bytes (Fig. 7).

In this experiment, the attack packets are generated by IXIA with randomly target IP and with the packet size of 64 Bytes to strengthen the attack effect. Assume that in IXIA simulation the attack flow sent to four switches respectively are $A1, A2, A3$ and $A4$, and attack flow rate is 200 Mbps$\times 4$ ($e.g.$ $A1 = A2 = A3 = A4 = 200$ Mbps). Without MTD defense, there is only one single controller at work and the total attack flow it receives is $A1 + A2 + A3 + A4$, which will increase the controller's CPU occupancy rate and degrade its performance. If MTD defense is initiated, the controller will give flow lead order to the switch for the latter to forward unmatched flow to Flood Filtering equipment and at the same time, notify the switches to randomly select a new controller (strategy $g_d(2)$ in Table 1.). At the beginning, there will be a time window for Flood Filtering equipment collecting route-info from controllers to make route-map. Only the hash of network address, not host address, will be used in route-map hash

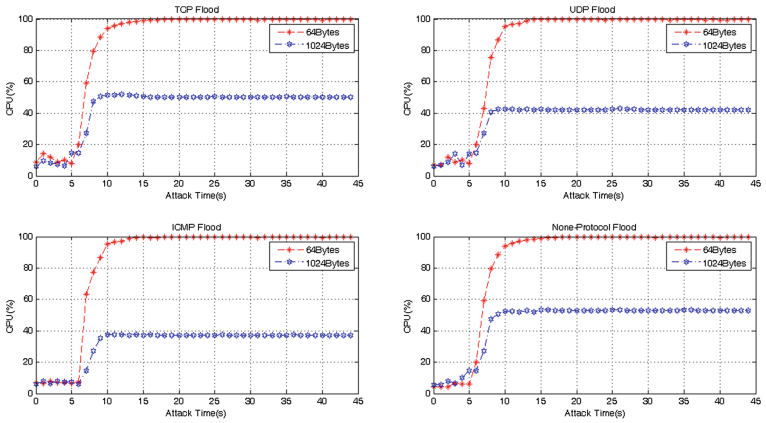

Fig. 7. Flood with different protocol

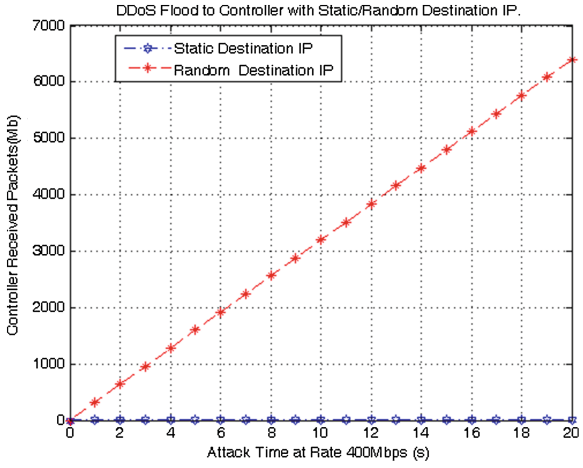

Fig. 8. DDoS flood to controller with static/random destination IP

table. When filtering the flood, the equipment just matches network address' hash in route-map. In this case, average data flow the controller receives will be decreased. According to statistical theory, the average attack flow for per controller will be F_a, where D is the attack flow the Flood Filtering Equipment drops.

$$F_a = \frac{1}{n}(\sum_{i=1}^{n} A_i - D) \tag{3}$$

In ideal conditions, if Flood-Filtering equipment can filter most of the attack flood, F_a is nearly equal to 0. Even if D is 0, which means Flood-Filtering equipment is not working, the value of F_a will be much smaller than that in the

case of single controller, which proves that MTD defense can effectively resist the harm of Blind DDoS attacks.

Figure 9 shows that Blind DDoS attack can destroy a single controller and increase its CPU occupancy rate to a very high value. And with MTD system, the number of controllers will increase and the packets received by one controller will decrease.

The experiment shows that MTD in SDN can effectively alleviate the damage to the controllers and switches caused by Blind DDoS attacks.

4.3 Security Analysis

This paper defines SDN as an open-closed hybrid system, which provides an important basis for the construction of an appropriate Moving Target Defense model defending Blind DDoS attack. The core idea of this defense model is to build a security defense system without detection, which can reduce risks of Blind DDoS in three attack kill chains, e.g. *Reconnaissance*, *Attack Launch* and *Persistence*.

Anti-Reconnaissance. Scanning plays an important role in the implementation of Blind DDoS attack. First, scanning can help identify whether the target network is SDN since Blind DDoS attack are only effective to SDN. Second, in order to make Blind DDoS attack more effective, scanning detection can be used to determine the range of random destination IP to make sure that its chance to match the flow entry is minimal.

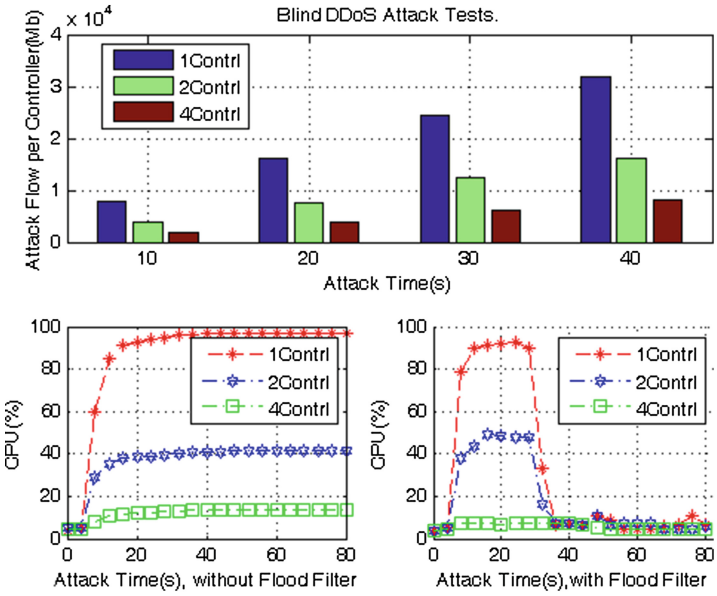

Fig. 9. MTD against blind DDoS attack

If the attacker wants to get useful information in reconnaissance, he should be able to distinguish the response times of First-Packet and Last-Packet. As moving target defense Randomly Delay strategy is adopted in our approach, the response times in scanning attacks will be indistinguishable. Since in our solution the Randomly Delay strategy is applied to every scanning packet, the $One - Time\ Padding$ method can be used to make a completely randomized sequence and the response time of two packets is statistically indistinguishable. In this way, it can effectively resist the effects of scanning attacks and play an active role in defending subsequent Blind DDoS flood.

Anti-Attack Launch. The MTD architecture proposed by this paper adopts a multi-controller pool, where the switch can shift the controllers randomly in the event of Blind DDoS attack. On the one hand, multi-controller can effectively alleviate the pressure of Blind DDoS attack on single controller; on the other hand, mobility of multi-controller will also increase the difficulty for the attacker to launch attacks directly on the controller, thus improves its security. Since the network between controller and switches is closed to attackers, there are enough IPv4 or IPv6 addresses for controllers. So the entropy of shifting IP space is big enough.

In addition to multi-controller strategy, this paper also presents a lightweight flood flow filtering method based on route map. Previous analysis shows that Blind DDoS attack is a special means of attack which requires the attacker to construct non-existent or random destination IP address in order to achieve best attack effects. In this paper, we gather the history record of routing tables on the controller as the basis of flood filtering, which is able to filter a large proportion of Blind DDoS attack quickly.

Anti-Persistence. Although there is little possibility for Blind DDoS attacks to install additional back-doors or access channels to keep persistence to the controller, it's not sure whether other kinds of attacks can do that, such as Blind Injection attacks or Buffer Overflow attacks. Besides anti-Blind DDoS, our MTD model with multi-controller can also reduce the persistence risks of Blind Injection attacks or Buffer Overflow attacks by randomly changing and refreshing controllers.

5 Discussion and Future Works

The above analysis demonstrates two key steps by which the attacker launches Blind DDoS attack on SDN controller, where the first one is scanning detection and the second one is sending of a large number of packets in abnormal structure, or attack packets with randomly destination IP address. In this paper, we construct a defense model and system based on MTD to cope with the Blind DDoS attack in SDN environment.

However, there are some limitations in this approach. On the one hand, in order to resist the scanning SDN attack, a method of random packet transmission delay is adopted, which will affect the normal data transfer performance. On the

other hand, Flood-Filtering equipment filters attack flow based on the history of routing information which requires prolonged keeping of routing tables, but how to synchronize route tables in multi controllers is not considered herein. By default, each controller will regenerate its own routing tables after shifted to online mode. This may produce false negatives because the routing tables may have expired.

To the first problem, we will research on sampling-delay method focusing on the high-speed, large volume of data transmission, while maintaining the low-speed transmission delay to every packet.

In order to solve the problem of false negatives to attacks, we will optimize the updating mechanism of route table to reduce the possibility of attacks by the attacker availing expired route tables. And another available scheme we can adopt is to replace simple route querying with SOM [27] and we also plan to adopt data mining methods to realize more accurate attack data stream filtering.

In spite that the model of randomly shifted controllers pool proposed in this paper is able to solve the problems of time delay and false negatives to attacks, it also has some limitations for it can only be used in Openflow1.3 and later versions. How to realize synchronization of multi controllers non-dependent on OpenFlow protocol version is worthy of further study.

6 Related Research

Although OpenFlow Specification White Paper [3] has proposed muti-controller since version 1.3, its application is still not clearly defined. OpenFlow classifies controllers into three kinds: *master*, *slave* and *equal*. However, as the configuration of mutli-controller is static and unable to be dynamically expanded, its security is at stake. To solve this problem, we give our approach using controllers pool instead of a single controller. Shin et al. [13] addresses the saturation challenge by the SYN Cookie. At low-rate [15,16] of TCP DDoS attack, SYN Cookie is a useful method to prevent flooding attack in SDN. But this approach will take an expensive computing resource in switch. When attack flow becomes very intensive, the switch's performance will slow down until it cannot work any longer. In our solution, we use MTD to select controllers randomly, so the flow in switch can just do matching action as usual without being interrupted. SYN flood [10,11] is just one type of those DDoS attacks. Any other flood, such as UDP flood, ICMP flood, etc., also can destroy SDN controller. Our defense system can resist more kinds of network protocol used by Blind DDoS attack. The literature [4] presented a method of identifying SDN architecture by comparing the system's response times to the same packets sent for two times in succession. Where it is a SDN network, DDoS attacks may be launched to consume resources of its control surfaces and forwarding surfaces.

Dixit et al. [22] proposed a solution named flexible distributed controller, which can dynamically increase or reduce the number of controllers by monitoring the load of controllers. Jafarian et al. [23] adopted OpenFlow to realize moving target defense. It differs from this paper in that, the object it defended is

the host in SDN, while that of this paper is SDN controller. In paper [13], SYN Cookie was adopted and the state of part SYN was represented by the switch to reduce the impact of DDoS attacks. The defect of this method lies in that it is just effective against SYN flood DDoS attacks and this solution requires changes in the switches' software programs and hardware programs, which is both costly and scarcely extensible. Shin et al. [13] also proposed a MTD method to defend brute force scanning. It can confuses the responding information to scanning attacks and can increase difficulty to attackers. Whereas, it is ineffective to Blind DDoS attacks and it is also ineffective to low rate scanning attack on SDN. The Crossfire attack [9] is not *Blind* because it require know and carefully select the links to the victims before launching attack.

The above literatures conduct researches on securities of hosts or controller in SDN [12] from the perspectives of applying SDN to security or vice versa. Our approach differs obviously from these methods in that we fist focus on defending Blind DDoS attacks based on MTD without detection.

7 Conclusion

SDN is new network architecture with immature technology and plenty of security risks. The security of SDN has become a focus of study in the field of network security. As controller is the core of SDN, SDN architecture with a single controller are vulnerable to performance bottlenecks and single point of failure. In this paper, we first propose the concept of Blind DDoS attack which is one of new threats to SDN. Then we analyze in details the principle of Blind DDoS attack, attack simulation and its harm, and proposed an attack defense method based on moving target defense. It proposes a novel MTD model to render the defender more effective and efficient. This method is advantageous as it adopted multi-controller, which is dynamically extensible with changes in attack flow. By randomly changing the packets delay in the switches, our approach can resist scanning attacks. Experiment and security argumentation demonstrate that this method is convenient to implement and can effectively defend Blind DDoS attack.

Acknowledgments. This paper is supported by the "Strategic Priority Research Program" of the Chinese Academy of Sciences, Grants No. XDA06010701, XDA06040502 and CAS Project No. XXH12501.

References

1. McKeown, N., Anderson, T., Balakrishnan, H., et al.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
2. http://www.OpenFlow.org/. Accessed December 2013
3. OpenFlow Switch Specification v1.3.0 (2013). https://www.open-networking.org/images/stories/downloads/specification/OpenFlow-spec-v1.3.0.pdf

4. Shin, S., Gu, G.: Attacking software-defined networks: a first feasibility study. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 165–166. ACM (2013)
5. Lau, F., Rubin, S.H., Smith, M.H., et al.: Distributed denial of service attacks. In: 2000 IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, pp. 2275–2280. IEEE (2000)
6. Curtis, A.R., Mogul, J.C., Tourrilhes, J., et al.: DevoFlow: scaling flow management for high-performance networks. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 254–265 (2011). (ACM)
7. Tootoonchian, A., Gorbunov, S., Ganjali, Y., et al.: On controller performance in software-defined networks. In: USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE) (2012)
8. Yu, M., Rexford, J., Freedman, M.J., et al.: Scalable flow-based networking with DIFANE. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 351–362 (2011)
9. Kang, M.S., Lee, S.B., Gligor, V.D.: The crossfire attack. In: 2013 IEEE Symposium on Security and Privacy (SP), pp. 127–141. IEEE (2013)
10. Wang, H., Zhang, D., Shin, K.G.: Detecting SYN flooding attacks. In: Proceedings of the IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, vol. 3, pp. 1530–1539. IEEE (2002)
11. Siris, V.A., Papagalou, F.: Application of anomaly detection algorithms for detecting SYN flooding attacks. Comput. Commun. **29**(9), 1433–1442 (2006)
12. Braga, R., Mota, E., Passito, A.: Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: 2010 IEEE 35th Conference on Local Computer Networks (LCN), pp. 408–415. IEEE (2010)
13. Seungwon, S., Yegneswaran, V., Porras, P., Gu, G.: AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. In: Proceedings 20th ACM Conference on Computer and Communications Security (CCS 2013), Berlin, Germany, November 2013
14. Floodlight. http://floodlight.openflowhub.org. Accessed December 2013
15. Sun, H., Lui, J.C.S., Yau, D.K.Y.: Defending against low-rate TCP attacks: dynamic detection and protection. In: Proceedings of the 12th IEEE International Conference on Network Protocols, 2004, ICNP 2004, pp. 196–205. IEEE (2004)
16. Kuzmanovic, A., Knightly, E.W.: Low-rate TCP-targeted denial of service attacks and counter strategies. IEEE/ACM Trans. Network (TON) **14**(4), 683–696 (2006)
17. Wang, F., Uppalli, R., Killian, C.: Analysis of techniques for building intrusion tolerant server systems. In: 2003 IEEE Military Communications Conference, 2003, MILCOM 2003, vol. 2, pp. 729–734. IEEE (2003)
18. Nguyen, Q.L., Sood, A.: A comparison of intrusion-tolerant system architectures. IEEE Secur. Priv. **9**(4), 24–31 (2011)
19. Nguyen, Q.L., Sood, A.: Designing SCIT architecture pattern in a cloud-based environment. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 123–128. IEEE (2011)
20. Hardman, O., Groat, S., Marchany, R., et al.: Optimizing a network layer moving target defense for specific system architectures. In: Proceedings of the ninth ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2013, pp. 117–118. IEEE Press (2013)
21. Jackson, T., Homescu, A., Crane, S., Larsen, P., Brunthaler, S., Franz, M.: Diversifying the software stack using randomized NOP insertion. In: Jajodia, S., Ghosh, A.K., Subrahmanian, V.S., Swarup, V., Wang, C., Sean Wang, X. (eds.) Moving Target Defense II. Application of Game Theory and Adversarial Modeling, vol. 100, pp. 151–173. Springer, New York (2013)

22. Dixit, A., Hao, F., Mukherjee, S., et al.: Towards an elastic distributed SDN controller. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 7–12. ACM (2013)
23. Jafarian, J.H., Al-Shaer, E., Duan, Q.: Openflow random host mutation: transparent moving target defense using software defined networking. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 127–132. ACM (2012)
24. Manadhata, P.K., Wing, J.M.: A formal model for a system's attack surface. In: Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Sean Wang, X. (eds.) Moving Target Defense. Creating Asymmetric Uncertainty for Cyber Threats, vol. 54, pp. 1–28. Springer, New York (2007)
25. Song, H.: Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 127–132. ACM (2013)
26. Feng, Y., Guo, R., Wang, D., Zhang, B.: Research on the active DDoS filtering algorithm based on IP flow. In: 2009 Fifth International Conference on Natural Computation, pp. 628–632. IEEE (2009)
27. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
28. Broder, A., Mitzenmacher, M.: Network applications of bloom filters: a survey. Internet Math. **1**(4), 485–509 (2004)