Area-Dividing Route Mutation in Moving Target Defense Based on SDN

Huiting Tan(✉), Chaojing Tang, Chen Zhang, and Shaolei Wang

National University of Defense Technology, Hunan 410072, China
Tanhuiting24@163.com

Abstract. To enhance mutation efficiency and proactively defend against denial of service attacks in moving target defense, we propose an effective and speedy multipath routing mutation approach called area-dividing random route mutation (ARRM). This approach can successfully resist denial of service attacks with acceptable CPU overhead and reduce convergence time caused by route mutation. Our contribution in this paper is threefold: (1) we provided model and method for smooth deployment of ARRM on software-defined networks; (2) we proposed extended shortest path calculation and route selection method to identify and select efficient route; (3) we simulated the interaction between ARRM defender and DoS attacker and develop analytical and experimental models to investigate the effectiveness and costs of ARRM under different mutation intervals and adversarial parameters. Our analysis and preliminary implementation show that ARRM can protect flow packets from being attacked against persistent DoS attackers and prolong attackers' response time. Moreover, compared with traditional RRM schemes, our implementation shows that ARRM can efficiently decrease the recalculation time delay caused by route mutation with acceptable CPU costs.

Keywords: Area-dividing route mutation · Moving target defense · Software defined networks

1 Introduction

With the massive use of Internet, we have acquired enormous benefits and convenience. However, we also suffered tremendous attacks and threats. Currently, networks are static, isomorphic and definite, which allows attackers to reconnoiter a system at leisure to investigate networks and explore vulnerabilities before attacking. Additionally, once they acquire a privilege, they can maintain it for a long time. A promising approach eliminating these asymmetric advantages is called moving target defense (MTD) [1], it changes various aspects of the network over time to shift the network's attack surface and make targets harder to "hit." While MTD is still in its infancy, this idea has gained significant attention in recent years with the increasing adoption of several enabling technologies such as software-defined networks (SDNs).

MTD techniques can be divided into five categories according to different levels. They are dynamic data, dynamic software, dynamic runtime environment, dynamic platform, and dynamic networks [2]. On network level, there are several techniques

Z. Yan et al. (Eds.): NSS 2017, LNCS 10394, pp. 565–574, 2017.
DOI: 10.1007/978-3-319-64701-2_43

which change network configurations and dynamic routing is one of them. In many protocols, the route selection is based on shortest path. The static route selection offers significant advantages for adversaries to eavesdrop and gather information or launch DoS attacks on certain network flows. In 2012, Ehab Al-Shaer, Qi Duan and Jafar Haadi Jafarian [3] proposed a moving target defense technique called Random Route Mutation (RRM). This is the first work to apply random route mutation in terms of multiple performance and security constraints. It presented algorithms to implement RRM technique and its simulation and preliminary implementation show that RRM is feasible and can defend eavesdropping and infrastructure DoS attacks effectively. In 2013, Duan Q, Al-Shaer E and Jafarian H [4] further investigated the feasibility of RRM in conventional network and develop implementation based on SDN. Its evaluation results show that RRM can effectively decrease the percentage of attacked packets caused by eavesdropping or DoS to less than 10% of the case of static routes.

However, mutated routes should be pre-calculated and staged in router configurations in advance but current route mutation methods have the disadvantages of high mutation costs, low efficiency and long processing delay. Once the scale of the network is getting larger, the convergence time of router is getting much longer.

The major contributions of our work as compared with former works include below:

* We proposed a new RRM-based model called area-dividing random route Mutation (ARRM) to deal with long convergence time problem. Previous works in RRM mainly focus on satisfying overlap constraint, capacity constraint and QoS constraint while ignoring the efficiency of route mutation. ARRM can decrease convergence time caused by link changes and enhance mutation efficiency.

* We provided an efficient and practical model to implement ARRM in SDN networks and evaluate ARRM effectiveness and overhead costs under different mutation interval and adversary parameters.

Our analysis, evaluation and experimentation show that ARRM can effectively defend against denial of service attacks with acceptable CPU costs while reducing the size of routing tables and reconfiguration time.

The remainder of this article is organized as follows. Section 2 presents related MTD techniques especially IP-hoping techniques. In Sect. 3, we illustrate the model and implementation of ARRM. Section 4 discusses the theoretical analysis of the ARRM method and section V illustrates the simulation evaluation results. Section 5 concludes our work and proposes efforts can be done in the future.

2 Related Works

There are several moving target defense techniques on network level and lots of ideas and concepts of random route mutation are based on them. In 2001, Kewley et al. [5] proposed the DyNAT approach which aims at disguising the characteristics of hosts in public domain of the network by a cooperative IP mutation scheme between a server and its clients. In 2003, Atighetchim et al. [6] came up with the APOD (Applications That Participate in Their Own Defense) scheme and use hopping tunnels based on address and port randomization to distinguish end parties between sniffers.

Another IP mutation approach called NASR [7] was proposed by Antonatos et al. in 2007. This method aims at defending against hitlist worms.

However, the three techniques above need to change the network configuration of the end host and bring in lots of overhead costs. To settle this problem, RHM (Random Host IP Mutation) [8] scheme was presented in 2011 by Ehab Al-Shaer et al. This approach turns end-hosts into untraceable moving targets by transparently mutating their IP addresses in an intelligent and unpredictable fashion and without sacrificing network integrity, performance or manageability. In RHM, moving target hosts are assigned several virtual IP addresses which change randomly over time. In order to prevent disruption of active connections, the IP address mutation is managed by network appliances and totally transparent to end-host. Based on the idea of RHM, Ehab Al-Shaer et al. [3] illustrated the concept of Random Route Mutation (RRM) in 2012 and defines algorithms which can achieve path randomization between a source and a destination.

3 Prototype Skeleton and Implementation of ARRM

We proposed Area-dividing Random Route Mutation (ARRM) and illustrate its architecture in Fig. 1. In an autonomy system, we divide the entire network into the backbone area 0 and several sub-areas. When host 1 in a sub-area A intends to interconnect with host 2 in another sub-area B, A's boundary switch should firstly connect with backbone switch in the backbone area. Then, backbone switch will interact with B's boundary switch and finally accomplish the connection between two sub-areas. With this design, when internal link states in an area are changed, controller only needs to update the routing table of internal switches and shortest path calculation can be done merely within the area instead of the entire network.

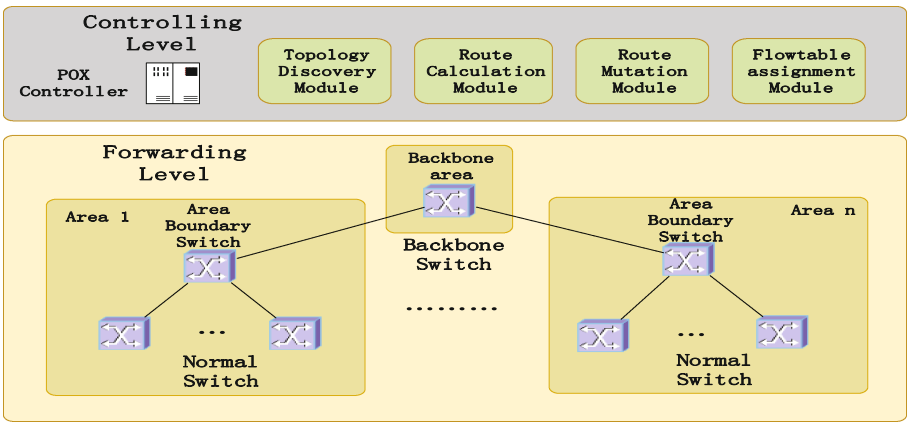

Fig. 1. Implementation structure of ARRM.

3.1 Openflow Switches Function Design

Openflow switches [9] are the core components of Openflow network and each Openflow switch contains 1 or more flow table, each flowtable is made up of multiple flow entries. Switches only forward corresponding flowtable while external controller implements MAC address learning, flowtable construction and maintenance, route mutation and et al.

According to the structure and function of system envisaged in Fig. 1, there are three different types of switches, normal switches, backbone switches and area boundary switches. Normal switches are located within the sub-areas and complete basic functions such as forwarding. Backbone switches are switches which locate in the backbone area. Area boundary switches are located on the area border and interconnect with other area boundary switch through the backbone switches.

Normal switches realize basic functions such as flowtable matching and forwarding. Once a switch receives a new packet, it will extract field parameters from the flow packet and match it with matching fields. Packets which belong to the same flow will be matched by the corresponding flow entry and follow its actions and update the counter. If cannot be matched, the packets will be forwarded to the controller and urge controller to decide forwarding ports and issue a new corresponding flowtable.

Backbone switches are located in the backbone area and share same basic functions with normal switches. Besides, they play the role of relay switches between two sub-areas and crossing-area actions need to go through backbone switches.

Area boundary switches also share basic functions with normal switches. All Area boundary switches carry two routing tables, one is regional routing table containing link information of its area and the other is cross-regional routing table containing information outside its area. Area boundary switches must be interconnected through backbone switches. When calculating the shortest distance between two hosts in different areas, the shortest path is combined of three separate paths and each separate path is calculated using Floyd-Warshall [10] algorithm. They are path P1 from source host to its area boundary switch, path P2 from source area boundary switch to destination area boundary switch, path P3 from destination area boundary switch to the destination host. Eventually, the crossing-area interconnection path is a combination of P1, P2 and P3.

3.2 Route Calculation Module

Based on the Floyd-Warshall algorithm, we proposed improved shortest path algorithm for both regional and cross-regional cases in route calculation module. The calculation can be done simply according to the source address, the destination address, and real-time network topology.

Hosts which are in the same area use Floyd-Warshall algorithm to calculate the shortest path to each other. When calculating the route across regions, we need to separately using Floyd-Warshall algorithm to calculate the shortest path P1from source host to the boundary switch and the shortest path P3from objective boundary switch to the destination host. Also, P2 which represents the path between two boundary

switches should be obtained. To simplify the model, we consider distance of P2 is a fixed value Cost2, while Cost1 and Cost3 separately represent minimum hop count of shortest path P1 and P3. Therefore, the shortest path crossing two region is a combination of these three separate paths and the final cost is Cost1 + Cost2 + Cost3.

When there is a link state change in a sub-area for crossing-area communication, unlike recalculating shortest paths of the entire network in traditional method, we only need to recalculate the shortest paths of the sub-area where a link change happened and the rest separate paths will stay the same. Therefore, the new shortest path will be combination of P1', P2 and P3. In this way, the recalculation time delay can be reduced and mutation efficiency is enhanced.

3.3 Route Mutation Module

Route mutation module complete the dynamic random routing. Its randomization is achieved in two ways. One is random choice among feasible routes, the other is random matching of switches' physical interfaces and logic addresses. We firstly obtain several feasible routes using Floyd-Warshall shortest path algorithm. Then, we construct a logic address pool when initializing the module. Facing specific data flow, we randomly select a logic address for each switch physical interface from address pool and establish the matching relationship. Selected logic addresses will be moved out from address pool to prevent distribution conflict. Actions above achieve the purpose of establishing dynamic random mapping between switch physical interface and logic address. In addition, we release all assigned logical addresses and initialize address pool periodically. The process of route mutation is clearly illustrated in Fig. 2.

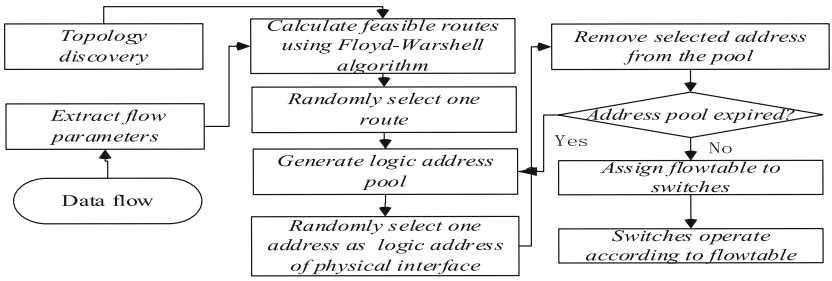

Fig. 2. Process of route mutation.

4 Theoretical Analysis

4.1 Performance Against Eavesdropping and Dos Attacks

Eavesdropping is a network layer attack that captures packets from the network transmitted by other computers and reading the data content in search of information. When there is no mutation in the case, attacker can eavesdrop one host's IP address and acquire the entire information. After bringing in route mutation, we disperse network

traffic by randomly choose one address from the logic address pool. This method puzzles the eavesdropper and attacker should eavesdrop all N_a possible IP address to make sure it get the full information as before. Therefore, proposed method can enhance the cost of eavesdropping.

Denial of service (DoS) attacks where an attacker utilizes massive machines to generate excessive traffic has caused severe service disruptions in recent years. Denial of service is typically accomplished by flooding the targeted machine or resource with tremendous requests to overload systems and prevent legitimate requests from being fulfilled. Various commercial solutions [11] addressing this problem have emerged. As for proposed method, since it brings uncertainty and changes into the network, it can also resist Dos attacks similarly.

Assuming T_0 is the requiring attacking time without route choice and address changes, N_a is the number of available logic address and N_r is the number of shortest routes. The requiring attacking time after bringing in address hopping and route mutation is

$$T = T_0 \left\{ 1 + \sum_{i=1}^{N_a N_r - 1} i \frac{C_{N_a N_r - 1}^{i}}{C_{N_a N_r}^{i} C_{N_a N_r - 1}^{i}} \right\} \tag{1}$$

From Eq. 1 we can see that proposed method increases attackers' time costs and requiring time increases as N_a and N_r increases.

We can further assuming N represents the number of Dos attacker, r represents attackers' attacking rate, τ represents the hopping gap, x represents the number of targeted packets, k is a constant value. Therefore, the average value $E(x)$ of x is

$$E(x) = \frac{kNr\tau}{N_a N_r} \tag{2}$$

Equation 2 shows that targeted packets number by DoS attackers decreases when available logic address and routes increases or hopping gap decreases.

4.2 Performance Against Internal Threats

Due to the facts that controlling platform and transferring platform are separated in SDN networks, even if the attackers are in the same network with the server and try to visit the target server, they still need to be checked by the SDN controller. Therefore, although the real IP address and port of sender and receiver have not changed through the whole process in our method, this method can still efficiently resist the internal threats and SDN controller here is used regarded as a filter gateway.

5 Implementation and Verification

We deployed ARRM and traditional RRM on a software-defined network (SDN) to verify our analytical model, evaluate their overhead and compare their performances. We use Mininet [12] python libraries as a topology generator and constitute a software-defined network. As for the controlling level, the network is managed by a python POX [13] controller.

5.1 Performance Against Eavesdropping and Dos Attacks

Firstly, we accomplished dynamic routing described in our structure and illustrated Traceroute [14] results in Fig. 3. As we can see in the graph, when we do the same Traceroute operation, the IP addresses of gateways change every time. The results demonstrates that we have successfully accomplished dynamic routing.

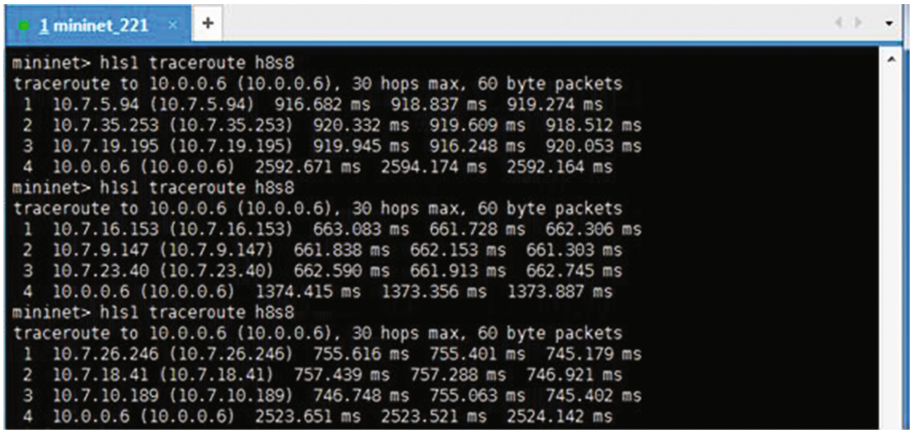

Fig. 3. Traceroute results of dynamic routing.

Then, to test system's ability of resisting DoS attacks, we launched typical DoS flooding attacks against specific destination host and illustrates the results in Fig. 4. As we can see in the figure, area-dividing route mutation method (given hopping gap is 10 s) can prolong the attacking time compared to system without area-dividing. Besides, as the attacking interval decreases, the difference among them enlarges.

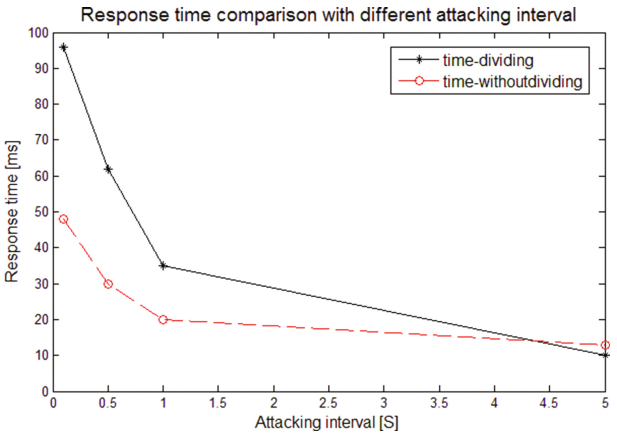

Fig. 4. Performance comparison with different Dos attacking intervals.

5.2 Traceroute Test

The system needs to recalculate the shortest path and distance every time when there is a change of link states. To measure the efficiency of recalculation, we use the 'Traceroute' instruction to clarify the time costs. We change the hopping gap to 5 s, 10 s, and 15 s and obtain the results seen in Fig. 5. Results indicate that route mutation method with area-dividing can decrease the recalculation time. Also, it indicates that decreasing the hopping gap can prolong recalculation time and the tendency is more obvious in system with area-dividing method.

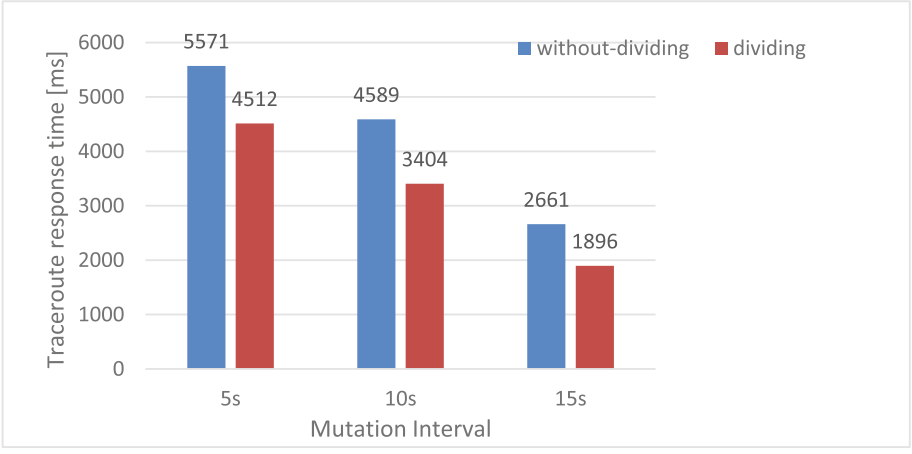

Fig. 5. Performance comparison with mutation intervals.

5.3 CPU Overhead Cost

To measure the extra CPU costs brought by the area-dividing route mutation in SDN controller, we simulate with different mutation interval to compare the CPU costs of traditional RRM and ARRM. Results in Fig. 6 shows that the maximum extra costs of ARRM is approximately 5.3% while the minimal deviation is around 1%. The difference is comparatively small and can be neglected. Therefore, we believe extra CPU cost brought by area-dividing is acceptable.

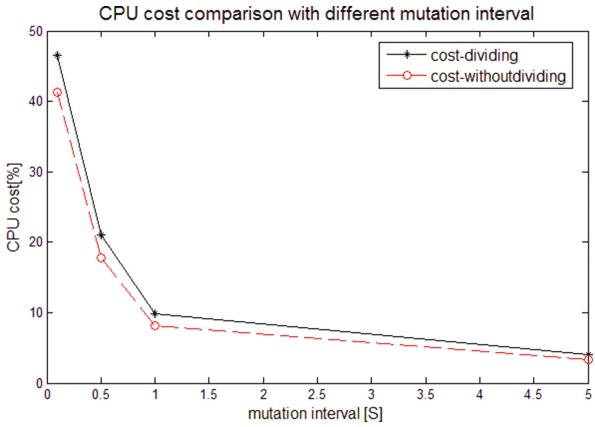

Fig. 6. CPU cost comparison with different mutation interval.

6 Conclusion and Future Works

To solve the problem that route recalculation and updating time rapidly grows with the network scale in current route mutation method, we proposed an area-dividing random route mutation (ARRM) technique. We let cross-regional communications be done through the backbone area. Therefore, routing table updating caused by link changes can be limited at borders of areas and significantly decrease the convergence time and enhance route mutation efficiency.

Our implementation of area-dividing route mutation technique is done on a SDN based network and several comparisons with traditional route mutation technique are illustrated. The results show that proposed method can decrease recalculation and updating time and enhance route mutation efficiency while maintaining acceptable CPU costs.

For future, considering the fact that regular mutation intervals can be easily detected and mastered by the attacker, we plan to randomize the mutation intervals and further increase uncertainty of the route mutation system. Also, it is complicated for a single controller to manage a large-scale network therefore we intend to introduce several controller into large-scale system.

References

1. NITRD CSIA Homepage, https://catalog.data.gov/dataset/trustworthy-cyberspace-strategic-plan-for-the-federal-cybersecurity-research-and-development, Accessed 27 May 2017
2. Zhuang, R., Deloach, S.A., Ou, X.: Towards a theory of moving target defense. In: 1st ACM Workshop on Moving Target Defense Proceedings, pp. 31–40. ACM, New York (2014)
3. CPS-VO Homepage, http://cps-vo.org/node/3854, Accessed 11 June 2017
4. Duan, Q., Al-Shaer, E., Jafarian, H.: Efficient random route mutation considering flow and network constraints. In: Communications and Network Security Proceedings, pp. 260–268. IEEE, National Harbor (2013)
5. Kewley, D., Fink, R., Lowry, J., et al.: Dynamic approaches to thwart adversary intelligence gathering. In: DARPA Information Survivability Conference & Exposition II, pp. 176–185. IEEE, Anaheim (2002)
6. Atighetchi, M., Pal, P., Jones, C.: Building auto-adaptive distributed applications: the QuO-APOD experience. In: International Conference on Distributed Computing Systems Workshop Proceedings, pp. 104–109. IEEE Computer Society, Washington, DC (2003)
7. Antonatos, S., Akritidis, P., Markatos, E.P.: Defending against hitlist worms using network address space randomization. Comput. Netw. Int. J. Comput. Telecommun. Netw. **51**(12), 3471–3490 (2007)
8. Al-Shaer, E., Duan, Q., Jafarian, J.H.: Random host mutation for moving target defense. In: Keromytis, Angelos D., Pietro, R. (eds.) SecureComm 2012. LNICSSITE, vol. 106, pp. 310–327. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36883-7_19
9. Mckeown, N., Anderson, T., Balakrishnan, H.: OpenFlow: enabling innovation in campus networks. Acm Sigcomm Comput. Commun. Rev. **38**(2), 69–74 (2008)
10. Hougardy, S.: The Floyd-Warshall algorithm on graphs with negative cycles. Inf. Process. Lett. **110**(8), 279–281 (2010)
11. Zhuang, R., Zhang, S., Deloach, S.A.: Simulation-based approaches to studying effectiveness of moving-target network defense. Nat. Symp. Moving Target Res. **53**(59), 15111–15126 (2013)
12. Kaur, K., Singh, J., Ghumman, N.S.: Mininet as software defined networking testing platform. In International Conference on Communication, Computing and Systems Proceedings (2014)
13. Shalimov, A., Zuikov, D., Zimarina, D.: Advanced study of SDN/OpenFlow controllers. In: Central & Eastern European Software Engineering Conference Proceedings, pp. 1–6. ACM New York (2013)
14. Augustin, B., Friedman, T., Teixeira, R.: Multipath tracing with Paris traceroute. In: End-to-End Monitoring Techniques and Services Proceedings, pp. 1–8. IEEE, Munich (2007)