Survey of software defined network and mutable network

Huai-Xi Wang*, Ni-Na Shu, Yong-Jie Wang and Chen Wang
Hefei Electronic Engineering Institute, 460 Huangshan Rd., Hefei, Anhui, 230031, P.R. China
E-mail: paper2submit@163.com
**Corresponding author*

Software defined network and mutable network are two main networking forms in these years. Traditional networking cannot meet the requirements of the renewal in information technology applications. We discuss mutable network and software defined network in this survey. We contrast two networking in motivations, crucial techniques and the practical scenarios.

Keywords: Cloud Computing; Moving Target Defense; Software Defined Network; Mutable Network.

1. Introduction

With the development of the information technology, some new types of network forms have been proposed and studied in industrial and academic areas. In these new networks, software-defined network and mutable network have attracted lots of attentions from researchers.

Software defined networking(SDN) [1-3] is a comprehensive term encompassing several kinds of network technology aimed at making the network as agile and flexible as the virtualized server and storage infrastructure of the modern data center. The goal of software defined network is to allow network engineers and administrators to respond quickly to changing business requirements. In a software defined network, a network administrator can shape traffic from a centralized control console without having to touch individual switches, and can deliver services to wherever they are needed in the network, without regard to what specific devices a server or other device is connected to. The key technologies are functional separation, network virtualization and automation through programmability.

Al-Shaer [4] proposed a new kind of dynamic and robust network in moving target defense [5-7] area, mutable network. Since then, mutable network has been studied and discussed from many aspects [8-10]. Mutable network enables networks to change their configurations such as IP address and routes randomly and dynamically while preserving the requirements and integrity of network operation. Mutable network can be used to hinder the adversary's capabilities in scanning or discovering network targets, launching DoS attacks and creating botnet structures.

2. Motivation

Software defined network and mutable network has been proposed in different motivations. Generally speaking, software defined network enables network administrators to manage network services through abstraction of higher-level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). This is very useful in network administration for cloud computing and big data center. Mutable network is mainly used to defend various cyber-attacks from Internet. The static nature of traditional network makes attackers own great advantages. Mutable network enables users to defend a system and increase the complexity of cyber-attacks by making the system less homogeneous, less static, and less deterministic.

2.1. *Software Defined Network*

Originally, software defined network focused solely on separation of the control plane of the network, which makes decisions about how packets should flow through the network from the data plane of the network, which actually moves packets from place to place. When a packet arrives at a switch in network, rules built into the switch's proprietary firmware tell the switch where to forward the packet. The switch sends every packet going to the same destination along the same path, and treats all the packets the exact same way. In a classic SDN scenario, rules for packet handling are sent to the switch from a controller, an application running on a server somewhere, and switches (data plane devices) query the controller for guidance as needed, and provide it with information about traffic they are handling. Controllers and switches communicate via a controller's "south bound" interface, usually OpenFlow, although other protocols exist.

Where a traditional network would use a specialized appliance such as a firewall or load balancer, an SDN deploys an application that uses the controller to manage data plane behavior. Applications talk to the controller via its "north-bound" interface. Up to the end of 2014, there is no formal standard for the application interface of the controller to match OpenFlow as a general south-bound interface. The OpenDaylight [11-13] controller's northbound API may emerge as a defacto standard over time, given its broad vendor support. The administrator can change any network switch's rules when necessary -- prioritizing, de-prioritizing or even blocking specific types of packets with a very granular level of control. This is especially helpful in a cloud computing multi-tenant architecture, because it allows the administrator to manage traffic loads in a flexible and more efficient manner. Essentially, this allows the administrator to use less expensive commodity switches and have more control over network traffic flow than ever before.

Software-defined networking is an approach to computer networking that allows network administrators to manage network services through abstraction of higher level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (control plane) from the underlying systems that forward traffic to the selected destination (data plane). SDN requires some method for the control plane to communicate with the data plane. One such mechanism, OpenFlow, is often misunderstood to be equivalent to SDN, but other mechanisms could also fit into the concept.

2.2. *Mutable Network*

Mutable network is used to defend a system and increase the complexity of cyber-attacks by making the system less homogeneous, less static, and less deterministic.

The network attack cycle spans number of steps including reconnaissance, finger printing, network mapping, exploitation, coordination, reporting, and propagation. In each step, the adversary relies on the static nature of cyber infrastructures to achieve the attack target effectively. The static nature of network configuration enables adversaries to discover and compromise network resources remotely. For instance, network configuration such as IP addresses, port numbers, platform type, service and patch version, protocols, service vulnerability and even firewall rules can be discovered using network scanning and fingerprinting tools. In addition, the accept-by-default Internet access

control makes network reconnaissance and zero-day worms inevitable. This calls for novel ideas to change the game of cyber security for the advantage of defenders in the face of ever advancing cyber-attacks. Mutable network attempts to archive this goal by using moving target defense techniques that force attackers to continuously chasing their target, deterring, and eliminating attacks without interrupting regular network traffic. This will eliminate the attacker's time and space advantage and create agility against advanced persistent threats.

As the network configuration is relatively static in the current game, the adversaries mainly rely on network reconnaissance (i.e., scanning and fingerprinting) as precursory step for lunching or propagating attacks in a network. Likewise, attack coordination assumes known and static host configuration. For instance, botnet communication and coordination depend on well-defined structure presuming static configuration such as IP address as well.

Mutable network is a new kind of network for moving target defense. This architecture allows for creating periodically alternative random configurations (called mutation), while maintaining the integrity and continuity of network operations and services. The frequently random mutation of hosts' location and identity in the network will constantly invalidate and deceive the adversary discoveries. The basic idea was initially proposed by Al-Shaer et.al. [14] in the "Moving Target" session during the National Cyber Leap Year Summit organized by DoD and NITRD.

3. Main Techniques

3.1. *Software Defined Network*

The main techniques in software defined network are as follows:

- SDN Application (SDN App): SDN Applications are programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN Controller via a northbound interface (NBI). In addition they may consume an abstracted view of the network for their internal decision making purposes. An SDN Application consists of one SDN Application Logic and one or more NBI Drivers. SDN Applications may themselves expose another layer of abstracted network control, thus offering one or more higher-level NBIs through respective NBI agents.
- SDN Controller: The SDN Controller is a logically centralized entity in charge of (i) translating the requirements from the SDN Application layer down to the SDN Datapaths and (ii) providing the SDN Applications with an

abstract view of the network (which may include statistics and events). An SDN Controller consists of one or more NBI Agents, the SDN Control Logic, and the Control to Data-Plane Interface (CDPI) driver. Definition as a logically centralized entity neither prescribes nor precludes implementation details such as the federation of multiple controllers, the hierarchical connection of controllers, communication interfaces between controllers, nor virtualization or slicing of network resources.

- SDN Datapath: The SDN Datapath is a logical network device that exposes visibility and uncontested control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources. An SDN Datapath comprises a CDPI agent and a set of one or more traffic forwarding engines and zero or more traffic processing functions. These engines and functions may include simple forwarding between the datapath's external interfaces or internal traffic processing or termination functions. One or more SDN Datapaths may be contained in a single (physical) network element—an integrated physical combination of communications resources, managed as a unit. An SDN Datapath may also be defined across multiple physical network elements. This logical definition neither prescribes nor precludes implementation details such as the logical to physical mapping, management of shared physical resources, virtualization or slicing of the SDN Datapath, interoperability with non-SDN networking, nor the data processing functionality, which can include OSI layer 4-7 functions.
- SDN Control to Data-Plane Interface (CDPI): The SDN CDPI is the interface defined between an SDN Controller and an SDN Datapath, which provides at least (i) programmatic control of all forwarding operations, (ii) capabilities advertisement, (iii) statistics reporting, and (iv) event notification. One value of SDN lies in the expectation that the CDPI is implemented in an open, vendor-neutral and interoperable way. SDN Northbound Interfaces (NBI) SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude). One value of SDN lies in the expectation that these interfaces are implemented in an open, vendor-neutral and interoperable way.

3.2. *Mutable Network*

The main techniques of mutable network [4] are as follows:

- Random Address Hopping: Network hosts will be frequently re-assigned random virtual IP address that will be used for routing independently form the actual IP address. The selection of the random IP addresses is synchronized across the network using crypto-based function and secret random keys to guarantee unpredictability and global configuration synchrony. The random IP addresses are selected from both the private address range and the unused address space which are sufficiently large. IPv6 offers much more available space for potential randomization. In this proposed approach, networked systems (i.e., end-hosts) will be assigned different addresses frequently based on random functions. One approach to achieve synchrony is to use round robin randomization.
- Random Finger Printing: Host responses will be intercepted and modified transparently to maximize the entropy in the system behavior and give a false OS and application identity. Without identifying the exact specifics of OS type and/or application servers, remotely exploitation will be infeasible. There are two mechanisms to randomize external system responses. One is to intercept and modify session control messages such as TCP 3-way handshake in order to cause false platform or service identification and deceive adversaries. Another technique is to use firewalls to deceive scanners by generating positive responses for all denies packets.
- Access Control Configuration based on Binary Decision Diagrams (BDDs): It results into a single Boolean expression (BDD) that represents all configuration interactions (i.e., flows and transformation) in the network. We will summarize our modeling discussion in the following sections.

4. Applications

4.1. *Software Defined Network*

SDN was commonly associated with the OpenFlow protocol [1] for remote communication with network plane elements since the latter's emergence in 2011. Since 2012, however, many companies have moved away from OpenFlow as a single-solution, and have embraced a number of different techniques. These include Cisco's Open Network Environment and Nicira's Network Virtualization Platform.

SDN architecture may enable, facilitate or enhance network-related security applications due to the controller's central view of the network, and its capacity to reprogram the data plane at any time. While security of SDN architecture itself remains an open question that has already been studied a couple of times in the research community, the following paragraphs only focus on the security applications made possible or revisited using SDN.

Several research works on SDN have already investigated security applications built upon the SDN controller, with different aims in mind. Distributed Denial of Service detection and mitigation as well as botnet and worm propagation, are some concrete use-cases of such applications: basically, the idea consists in periodically collecting network statistics from the forwarding plane of the network in a standardized manner (e.g. using Openflow), and then apply classification algorithms on those statistics in order to detect any network anomalies. If an anomaly is detected, the application instructs the controller how to reprogram the data plane in order to mitigate it.

Another kind of security applications leverages the SDN controller by implementing some moving target defense (MTD) algorithms. MTD algorithms are typically used to make any attack on a given system or network more difficult than usual by periodically hiding or changing key properties of that system or network. In traditional networks, implementing MTD algorithms is not a trivial task since it is difficult to build a central authority able of determining – for each part of the system to be protected - which key properties are hid or changed. In an SDN network, such tasks become more straightforward thanks to the centrality of the controller. One application can for example periodically assign virtual IPs to hosts within the network, and the mapping virtual IP/real IP is then performed by the controller. Another application can simulate some fake opened/closed/filtered ports on random hosts in the network in order to add significant noise during reconnaissance phase (e.g. scanning) performed by an attacker.

Additional value regarding security in SDN enabled networks can also be gained using FlowVisor [15] and FlowChecker [16] respectively. The former tries to use a single hardware forwarding plane sharing multiple separated logical networks. Following this approach the same hardware resources can be used for production and development purposes as well as separating monitoring, configuration and internet traffic, where each scenario can have its own logical topology which is called slice. In conjunction with this approach FlowChecker

realizes the validation of new OpenFlow rules that are deployed by users using their own slice.

Developing applications for software defined networks requires comprehensive checks of possible programming errors. Since SDN controller applications are mostly deployed in large scale scenarios a programming model checking solution requires scalability. These functionalities are provided among others through NICE.

Introducing overarching security architecture requires a comprehensive and protracted approach to SDN. Since it was introduced, designers are looking at possible ways to secure SDN that do not compromise scalability. This architecture is called SN-SECA (SDN+NFV) Security Architecture.

Nowadays, many cloud computing infrastructures adopt SDN as the key network, such as Google, Amazon, AliCloud, QingCloud. More and more hardware switch support SDN networking appear on the market.

4.2. *Mutable Network*

This technique could have varying levels of impact on an attacker depending on what the attacker is trying to accomplish. If an attacker is trying to disrupt the network, it could make flooding attacks more difficult if they did not have access to DNS addresses. Since this technique is not supposed to disrupt active connections, if an attacker can collect the information they need before a switch and establish a connection to the system, they may not be as impacted as much by this technique. A technique like this could have impact on applications or services that require constant connections or could disrupt current connections. More importantly, the protection offered is very limited. It does not protect against client-side attacks. The technique can also have severe scalability issues.

Mutable network attracts many attentions in academic area. It is a research idea and was not implemented sufficiently. Zhuang et.al [8] made experiments based on mutable network and verified its effect to defend cyber-attacks.

5. Conclusion

As the main networking forms in these years, software defined network and mutable network bring new ideas to IT industry. SDN would be a core component in cloud computing and big data center in the future. Meanwhile, mutable network would be an important measure to secure the information assets.

References

1. Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru M. Parulkar, Larry L. Peterson, Jennifer Rexford, Scott Shenker, and Jonathan S. Turner. Openflow: enabling innovation in campus networks. Computer Communication Review, 38(2):69–74, 2008.
2. Jim Esch. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1):10–13, 2015.
3. Diego Kreutz, Fernando M. V. Ramos, Paulo Jorge Esteves Ver íssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1):14–76, 2015.
4. Ehab Al-Shaer. Toward network configuration randomization for moving target defense. In Sushil Jajodia, Anup K. Ghosh, Vipin Swarup, Cliff Wang, and Xiaoyang Sean Wang, editors, Moving Target Defense - Creating Asymmetric Uncertainty for Cyber Threats, volume 54 of Advances in Information Security, pages 153–159. Springer, 2011.
5. Hamed Okhravi, MA Rabe, TJ Mayberry, WG Leonard, TR Hobson, D Bigelow, and WW Streilein. Survey of cyber moving target techniques. Technical report, DTIC Document, 2013.
6. David Evans, Anh Nguyen-Tuong, and John Knight. Effectiveness of moving target defenses. In Moving Target Defense, pages 29–48. Springer, 2011.
7. Ang Cui and Salvatore J Stolfo. Symbiotes and defensive mutualism: Moving target defense. In Moving Target Defense, pages 99–108. Springer, 2011.
8. Rui Zhuang, Su Zhang, Alex Bardas, Scott A DeLoach, Xinming Ou, and Achintya Singhal. Investigating the application of moving target defenses to network security. In Resilient Control Systems (ISRCS), 2013 6th International Symposium on, pages 162–169. IEEE, 2013.
9. Zhiming Wang, Jiangxing Wu, Guozhen Cheng, and Yiming Jiang. Mutine: A mutable virtual network embedding with game-theoretic stochastic routing. In 2015 IEEE Global Communications Conference, GLOBECOM 2015, San Diego, CA, USA, December 6-10, 2015, pages 1–6. IEEE, 2015.
10. Doran R Smestad Craig A Shue Marc Green, Douglas C Macfarland. Characterizing network-based moving target defenses. 2015.

11. Jan Medved, Robert Varga, Anton Tkacik, and Ken Gray. Opendaylight: Towards a model-driven SDN controller architecture. In Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2014, Sydney, Australia, June 19, 2014, pages 1–6. IEEE Computer Society, 2014.
12. Zuhran Khan Khattak, Muhammad Awais, and Adnan Iqbal. Performance evaluation of opendaylight SDN controller. In 20th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2014, Hsinchu, Taiwan, December 16-19, 2014, pages 671–676. IEEE Computer Society, 2014.
13. OpenDaylight. Opendaylight performance: A practical empirical guide. 2016. https://www.opendaylight.org/resources/odl-performance.
14. 14. Ehab Al-Shaer, Wilfredo Marrero, Adel El-Atawy, and Khalid Elbadawi. Network configuration in A box: Towards end-to-end verification of network reachability and security. In Proceedings of the 17th annual IEEE International Conference on Network Protocols, 2009. ICNP 2009, Princeton, NJ, USA, 13-16 October 2009, pages 123–132. IEEE Computer Society, 2009.
15. Rob Sherwood, Glen Gibb, Kok kiong Yap, Martin Casado, Nick Mckeown, and Guru Parulkar. Flowvisor: A network virtualization layer. Technical report, 2009.
16. Ehab Al-Shaer and Saeed Al-Haj. Flowchecker: configuration analysis and verification of federated openflow infrastructures. In 3rd ACM Workshop on Assurable and Usable Security Configuration, SafeConfig 2010, Chicago, IL, USA, October 4, 2010, pages 37–44. ACM, 2010.