INMTD: Intent-based Moving Target Defense Framework using Software Defined Networks

Muhammad Faraz Hyder
Department of Software Engineering
NED University of Engineering & Technology
Karachi, Pakistan
farazh@neduet.edu.pk

Muhammad Ali Ismail
Computer & Information Systems Engineering
NED University of Engineering & Technology
Karachi, Pakistan
maismail@neduet.edu.pk

Abstract—**Intent-Based Networking (IBN) is an emerging networking paradigm while Moving Target Defense (MTD) is an active security technique. In this paper, the Intent-based Moving Target Defense (INMTD) framework using Software Defined Networks is proposed. INMTD is the first effort in exploiting IBN for the design of an efficient Moving Target Defense (MTD) framework. INMTD uses the concept of shadow servers in order to counter the first stage of cyber-attacks, i.e. reconnaissance attacks targeted against servers running in SDN networks. INMTD comprises of an MTD application running on an SDN controller. The MTD application has reconnaissance detection, MTD movement, and MTD monitoring modules. The MTD application is integrated with the intent-based northbound API of SDN controller. INMTD not only provides protection against probing attacks, but it also provides high availability due to shadow servers. The proposed framework was implemented using Mininet and ONOS SDN controller. The proposed framework was assessed in terms of defender cost, attacker's effort, and introduced complexity in the system. The results substantiate the efficient protection against reconnaissance attacks at lower computational cost.**

Keywords-cyber kill chain; intent-based networking; moving target defense; software defined networks; SDN security

I. INTRODUCTION

Nowadays, cyber security is of critical importance. Moving Target Defense (MTD) is becoming one of the popular techniques, for providing active cybersecurity [1]. MTD makes the systems dynamic by constantly changing the attack surface, making it hard to predict and attack. MTD equalizes the cyber security field for defender and attacker by eliminating the advantages of the attackers. MTD has changed the concept of cyber defense since its first announcement in 2009 [2]. MTD constantly changes the attack surface to reduce the advantage of time of attackers. MTD changes the attributes periodically, for example, ports, IPs, so that the attacker cannot gain knowledge of the attribute through which attack can be launched. The change can be of two types: movement or transformation. MTD can be divided into numerous categories which can be chosen according to the required difficulty level for attackers [3]. Intent based networking (IBN) is an emerging networking paradigm [4]. In IBN, users define their applications' network requirements through policy. These policy instructions are referred as intents. IBN can also be used to fulfil the dynamic security requirements. Open Networking Foundation (ONF) has taken the initial steps towards its regularization of intent based networking. ONF provides recommendations for creating intent based North Bound Interfaces (NBI) [5]. Software Defined Networking (SDN) has recently gained substantial popularity as a networking paradigm. It primarily segregates the Control and Data planes [6]. Its architecture comprises of three fundamental layers namely Application, Control, and Data planes. Due to its dynamic nature and centralized control, numerous security applications can be implemented through it. SDN based MTD is an active area of research, while IBN is gaining popularity in the research community. However, to the best of our knowledge, no previous work has used IBN for an MTD solution.

In this paper, Open Network Operating System (ONOS) SDN controller based intent framework [7] was used. Various extensively used SDN controllers have similar types of intent based NBIs. The ONOS intent based framework receives intent instructions and converts them into a compilation form and then installs the intent [7]. These installations perform the required operations on the network. The intent can be withdrawn as per desire if it is no more required by an application. In this paper, an MTD solution is proposed using IBN and SDN. The notion is to exploit IBN for creating MTD based upon SDN. The proposed INMTD framework protects against the reconnaissance attack which is the first stage of any cyber-attack [8]. INMTD detects the reconnaissance traffic directed against the web server and redirects it towards the shadow servers using the intent based framework of ONOS controller. The proposed solution has low computational cost, high availability, and efficient redirections among its advantages. For the implementation of INMTD, Mininet emulator and ONOS Controller [9] were used.

II. RELATED WORK

An SDN based programming framework termed as Open Software Defined Framework (OSDF) was proposed in [10]. Network administrators mention their network requirements for each application using Application Manager Interface (API). Numerous network operations including setting up standard quality of network services, network configuration, and

Corresponding author: Muhammad Faraz Hyder monitoring can be managed through OSDF services. It also contains conflicting policy resolver. In a multilayer data center environment, IBN is implemented through virtualization abstraction networking [11]. For end-to-end service management, intent based reference architecture is proposed in [12]. The architecture is verified on OpenFlow and IoT based SDN testbeds and testing has been done in various domains. In [13], another intent based architecture is presented which facilitates automatic intent implementation in secure multilayer networks. The approach was also certified through testing on commercial testbeds. In [14], a reactive configuration using extended Intent-based Network Modeling (NEMO) language has been proposed. The reactive scheme will alter the network configuration automatically according to the shift in external environment. The change is representing the administrator's intent. The routing paths are shifting through bandwidth utilization in said scenario. An approach for business networks is proposed in [15] that implements IBN.

MTD architecture,developed in [16] using OpenFlow, alters the IP addresses randomly. This technique is named OpenFlow Random Host Mutation (OFRHM). The process of IP address alteration is hidden from users. The system is developed to use MTD against scanning. A collaborative mutation strategy named Network Moving Target Defense Technique based on Collaborative Mutation (TCM) is proposed in [17]. The combination of end-point mutation and routing mutation is set up which increases mutation space and reduces irregularity. Fingerprinting based mutation collision avoidance mechanism is also used to circumvent mutation collision. According to authors, TCM is more competent as compared to OFRHM and other similar techniques. A protective MTD mechanism for cloud networks is developed in [18]. The protection was done through the scheme of port hopping. A scoring strategy was used to check which cloud services are at risk. The score was measured by PageRank algorithm. MTD decisions were based on the vulnerability information obtained from the score. The impact of this MTD solution is more noticeable in large cloud networks than small scale cloud networks. An SDN-based MTD system named CHAOS was proposed in [19]. The system mystifies only the unexpected traffic without disturbing usual traffic. This is done by obfuscating each with a diverse level of security. SDN based MTD was proposed in [20] for throttling finger printing attacks which are targeting towards collecting operating system information. The proposed model was termed as FPH (fingerprinting hopping). FPH utilizes a game theoretic approach for constructing the optimal strategy for MTD. FRVM is a SDN based MTD framework [21]. The model derived its name based upon the multiplexing of virtual IP addresses. FRVM multiplexed virtual IPs based upon random fashion. In [22], a model was proposed for creating virtual topologies using SDN for protecting the reconnaissance attacks. The proposed framework utilizes the statistical information for potentially malicious nodes responsible for generating the probing traffic. In [23], the authors discussed the Distributed Denial of Services (DDoS) attacks on SDN networks. Their work also highlighted the anomaly detection techniques for SDN. The authors emphasized that the central plane of SDN is a lucrative target of attackers. The challenges with respect to the adaptation of cloud computing environment by telecom operators were addressed in [24]. The work is targeted towards specific country requirements. However, it can be extended for different countries.

III. Methodology

In this section, INMTD methodology is discussed in detail.

A. Threat Model

The attackers can be directly or indirectly connected to the SDN network. They can run different networking probing attacks against the different servers connected at the data plane. For this paper, the attacker's targets are the running web servers. As the first step of a cyber-kill chain, the attacker will attempt a reconnaissance attack. Each unique IP address is considered as an attacker. Each attacker can run up to 10 concurrent reconnaissance probes at a time. This will ensure a realistic probing frequency.

B. Proposed Model

The proposed framework comprises of an MTD application running in the Control plane. This MTD application utilizes the intent-based framework of ONOS Controller [9] in order to create MTD effect. Figure 1 represents the overall architecture of INMTD and its core components. The core component of MTD application is the reconnaissance detection module (RDM). It will detect any reconnaissance traffic directed towards web servers. This module is fundamentally implemented using SNORT [25] which is an open source IDS. The SNORT [25] code was modified in order to detect the reconnaissance traffic targeted towards the web server and then redirect the traffic towards the shadow web servers. The other important module of MTD application is the decision/movement strategy. As its name suggest its role is deciding the movement technique and frequency of the proposed MTD.

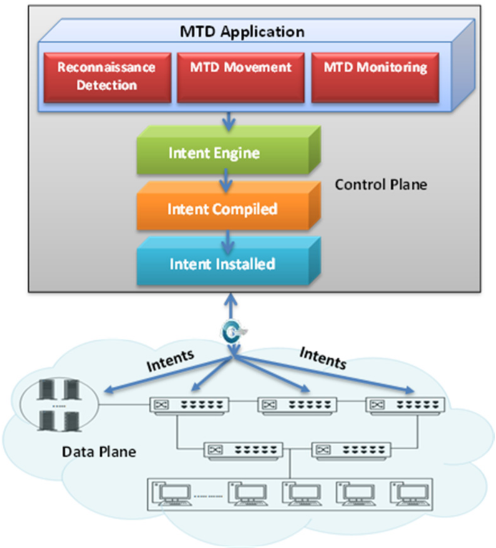

Fig. 1. INMTD architecture

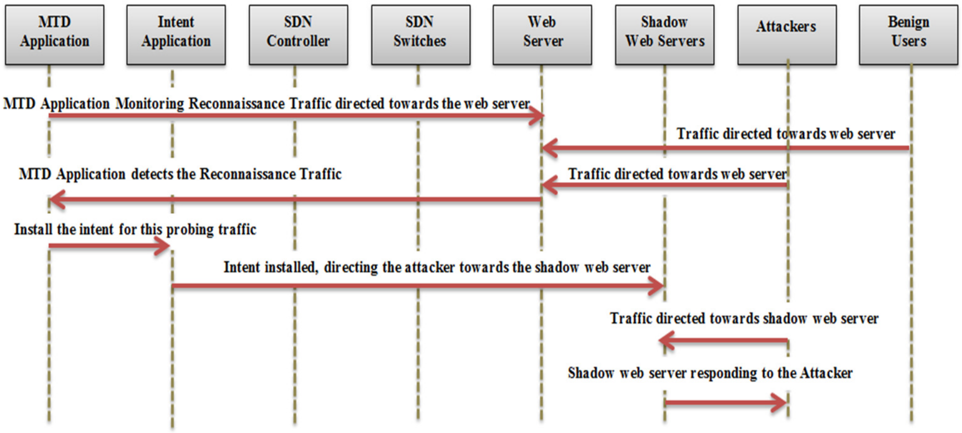
MTD Application
Intent Application
SDN Controller
SDN Switches
Web Server
Shadow Web Servers
Attackers
Benign Users
MTD Application Monitoring Reconnaissance Traffic directed towards the web server
Traffic directed towards web server
MTD Application detects the Reconnaissance Traffic
Traffic directed towards web server
Install the intent for this probing traffic
Intent installed, directing the attacker towards the shadow web server
Traffic directed towards shadow web server
Shadow web server responding to the Attacker

Fig. 2. INMTD workflow

The MTD monitoring module is responsible for the monitoring of the overall MTD system. The MTD application runs on the top of the ONOS Northbound Intent API [9]. This interface comprises of three parts: intent engine, compilation module, and the intent installation part. MTD application forwards the decision to the intent engine which performs the intent compilation through the intent compilation module. The compiled intents are finally installed in the required switch using the intent installation module. The installed intents will create flows in the switches. The fundamental advantage of INMTD is its effectiveness against probing traffic. The detection of probing traffic is difficult and there are false positive and false negative chances. Our approach redirects the traffic to the shadow web servers. The shadow web servers are replicas of the original web servers. Therefore, even if the RDM detects a legitimate user as an attacker, it still provides the web content to the user. Algorithm 1 represents the probing traffic detection and redirection through intent modification. This algorithm detects the reconnaissance traffic through matching the source and destination IP and port addresses and reconnaissance frequency.

Algorithm 1: Reconnaissance traffic detection & traffic redirection through intent modification

```
[Initialization of SDN based Network having different servers enabled intent based applications]
Initialization of RDM
function PacketArrival (srcIP, srcPort, dstIP, dstPort)
    if (dstIP = = WebServerIP AND Port=WebServer_port AND srcIP==MaliciousIP
        ##Possibility of Reconnaissance Attack on Web Servers
        RDM → Intent_App(Install_Intent)
        Intent_App → Modify the dest_IP using "setIpDst"
        SelectedShadowWebserver == RoundRobinSelection (list of K shadow webservers)
        SelectedShadowWebserver_setIpSrc == IP_address_Web_Server
        Shadow_Webserver → Attacker
    endif
    else
        Normal_SDN_Forwarding ()
end function
```

Once probing traffic is detected, it will be redirected towards the shadow web server while modifying the destination address of the server as one of the shadow servers. These shadow servers will be selected by the round robin fashion. The selected shadow server then responds to the probing traffic of the attacker. While responding to the probing traffic, the IP address of the shadow web server will be modified using the intent parameter of "setIpSrc" to match the IP address of the original web server. The attacker will actually conceive that it is connected to the original web server, while actually it is connected to the shadow server. This way a moving target defense effect will be created. The overall flow of INMTD is presented in Figure 2. MTD and intent based applications are running in the control plane. The RDM of INMTD is constantly monitoring the Data plane for any reconnaissance traffic directed towards the web servers. There are two types of users, benign and attackers. Benign users' traffic will follow the normal SDN forwarding mechanism. The traffic from the attackers will be detected and directed towards the shadow web servers by the MTD application using intent based application. An example of intents that are installed for directing the traffic against web server to the shadow server and prepare a response to look like generated from the original web server is this:

```
add-host-intent --ipSrc [IP Address of attacker] --ipDst [IP Address of web server] --tcpDst 80 --setIpDst [IP Address of one the shadow server]
add-host-intent --setIpSrc [IP Address Web Server] --setIpDst[IP Address of attacker
```

C. Experimental Setup

For the deployment of experimental setup, a Dell server with Intel Xeon CPU E5-2620 2.1GHz with 32 cores and 32GB RAM was used. Mininet [26] and ONOS Controller [9] were used for the creation of the SDN topology. Snort [25] was deployed as an IDS (Intrusion Detection System) mode. Nmap [27] was used for generating reconnaissance traffic. For the experimental analysis, the ONOS reactive forwarding application [9] was disabled. The reason is that only intent based forwarding was required. Next, intents were inserted based upon the probing traffic. Figure 3 represents the simulation setup for our proposed INMTD framework.

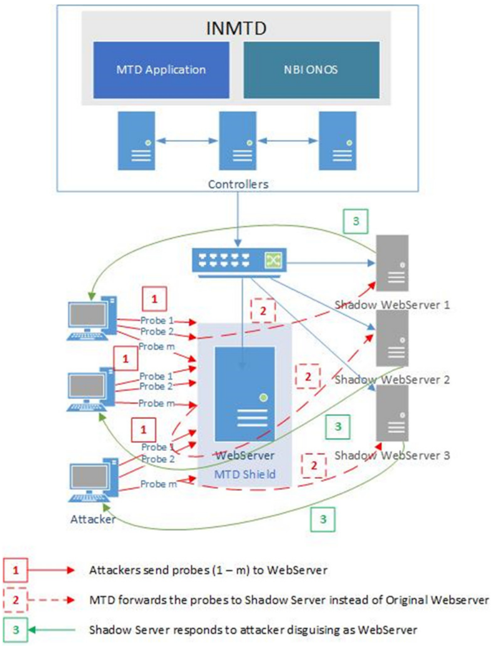
INMTD
MTD Application
NBI ONOS
Controllers
3
1
2
Probe 1
Probe 2
Probe m
Shadow WebServer 1
Shadow WebServer 2
Shadow WebServer 3
WebServer
MTD Shield
Attacker
1 Attackers send probes (1 – m) to WebServer
2 MTD forwards the probes to Shadow Server instead of Original Webserver
3 Shadow Server responds to attacker disguising as WebServer

Fig. 3. Simulation setup of INMTD

IV. RESULTS AND DISCUSSION

For Data plane security analysis, the case of one web server was considered. For one web server, there are k running shadow web servers. For the experimental analysis, we consider k=3. Different numbers of scans were performed in order to evaluate the performance of INMTD. The number of scans ranges from 100 to 3200. Each attacker can generate up to 10 scans. Each distinct IP address belongs to an attacker. This number of scans is realistic because increasing the number of scans beyond this limit will cause the IDS or the firewall system to permanently block the attacker's IP address. Table I presents the overall results of experiments.

TABLE I. RESULTS AFTER INMTD

No. of Scans	No. of Attackers	Successful redirections	Successful redirections (%)	No. of failed redirections	No. of intents	No. of flows	No. of blocked IP addresses
100	10	78	78.00	22	10	40	10
200	20	159	79.50	41	20	80	20
400	40	321	80.25	79	40	160	40
800	80	659	82.38	141	80	320	80
1600	160	1359	84.94	241	160	640	160
3200	320	2767	86.47	433	320	1280	320

A. Attacker Cost

The fundamental goal of MTD is to increase the attacker's effort. Attacker's cost primarily comprises of the number of scans performed while accurately detecting the platforms of the web server and port addresses, etc. Table II presents the attacker success for different numbers of scans for the cases of native SDN and INMTD enabled SDN. As evident from Figure 4, attacker's success substantially decreased after incorporating INMTD. Attacker's scanning attack success is around 97% to 98% against native SDN using Nmap tool [27]. For the current analysis, there are 100 to 3200 scans performed against a native SDN environment without any protection available. The success rate of attacker ranges from 97% to 98%. However, attacker's success reduced substantially when adopting the proposed INMTD. For 100 scans, the attacker success rate was 22.6% and it further reduced to 21% for 200 scans. In a similar fashion, the attacker success reduced as the number of scans increased, becoming 14% for 3200 scans. This is a substantial decrease in attacker scanning success rate.

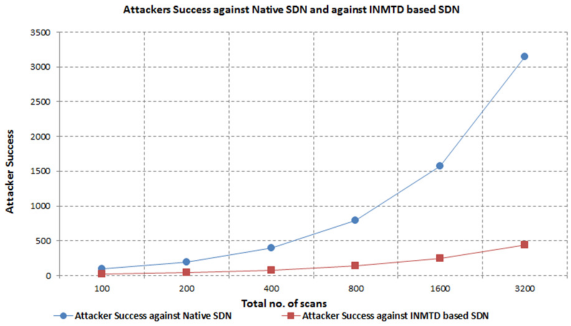

Fig. 4. Attacker success with and without INMTD

TABLE II. ATTACKER'S SCANNING SUCCESS AGAINST NATIVE AND INMTD BASED SDN

Total scans	Attacker success against native SDN	Attacker success against INMTD based SDN
100	97	22
200	195	41
400	391	79
800	788	141
1600	1569	241
3200	3150	433

B. Defender Cost

Defender's cost primarily comprises of the intent installation, IDS detection, and shadow web servers. Generally IDS is a part of any enterprise network. Moreover, generally a web application runs on multiple web servers. Therefore, the main cost is related to the intent compilation and installation. For this purpose we have calculated the number of flows injected for attacker's probing traffic with and without intents as presented in Table III. As mentioned above, each attacker can run 10 concurrent probs. Therefore, for each new IP address there will be a flow injected in the switches. It is clear from Table III that there is a slight increase in the number of flows, approximately 20% on average due to the addition of intents. Figure 5 presents the graph of the number of flows inserted in the switch for probing traffic with and without intents. As evident form the Table, for 100 scans, the number of flows before the intents' addition was 20. Afterwards, the number of flows increased to 22. Similarly, for 200 scans, there are 40 flows without intents. The number of flows extended up to 46 after adding the intents. For 3200 scans, there are 640 flows without intents, which increased to 788 after the intents were added. Therefore, the average intents increased the number of flows in the SDN switches approximately by 20%.

Hence, INMTD framework slightly increased the defender cost in terms of number of flows. Table IV and Figure 6 present the performance of INMTD for different numbers of scans. It is evident that INMTD successfully defended against probing attacks with accuracy from 78% to 86.5% for from 100 to 3200 scans. For 100 scans, the defender is capable of redirecting the probing traffic with accuracy of 78%. For 200 scans, the accuracy of INMTD increased to 79.5%. As the number of scans increased to 3200, the INMTD success reaches 86.47%.

TABLE III. NUMBER OF FLOWS WITH AND WITHOUT INTENTS

Total scans	Flows (after intents)	Flows
100	22	20
200	46	40
400	94	80
800	192	160
1600	390	320
3200	788	640

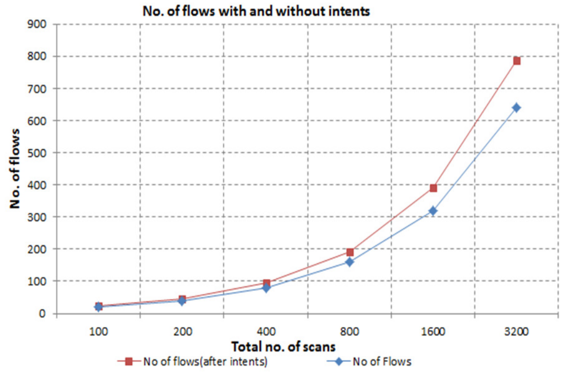

Fig. 5. Number of flows with and without intents

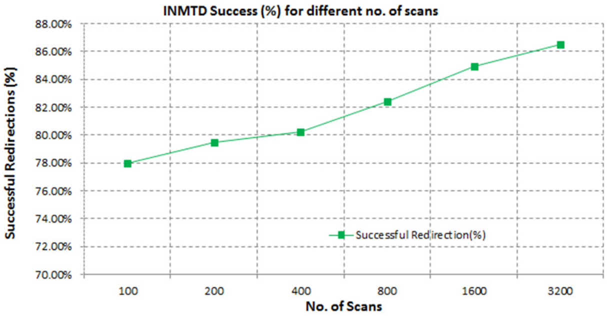

Fig. 6. INMTD success (%) for different scan numbers

TABLE IV. DEFENDER'S SUCCESS

Total scans	Successful redirections (%)	Defender success
100	78.00%	78
200	79.50%	159
400	80.25%	321
800	82.38%	659
1600	84.94%	1359
3200	86.47%	2767

C. Comparative Analysis of INMTD with other SDN based MTD

One of the novel contributions of the current work is the utilization of IBN for the design of SDN based MTD solution. To the best of our knowledge, no previous work has used IBN for the design of SDN based MTD solutions. Another critical advantage of INMTD is the distributed Control plane for higher availability. Figure 7 represents the comparative analysis of the proposed INMTD with three other well-known SDN based MTD solutions, namely OF-RHM [16], TCM [17], and FRVM [21]. For the purpose of comparative analysis, the proposed INMTD model and the other models were analyzed on the basis of successful redirection for the reconnaissance traffic and computation cost. The computational cost is determined in terms of number of flows injected in the SDN devices after the adaptation of the protection mechanism. The number of scans ranges from 100 to 3200. For theses scans, computational cost and successful redirections were calculated for the proposed INMTD, and the existing solutions. As indicated in Figure 7, INMTD achieves successful defense rate up to 86.5% with a computational cost of around 23%, while OF-RHM [16] achieved a success rate of 74.4% with 28.7% increase in computational cost. TCM [17] provided a success rate of 79.5% with 27.4% increase in computational cost. FRVM [21] had 83.2% success rate with 24.5% increase of the cost. INMTD has the highest success rate in comparison to the other three models. Moreover, INMTD computational cost is lower than OF-RHM [16] and TCM [17]. Its computational cost is almost similar to that of FRVM [21].

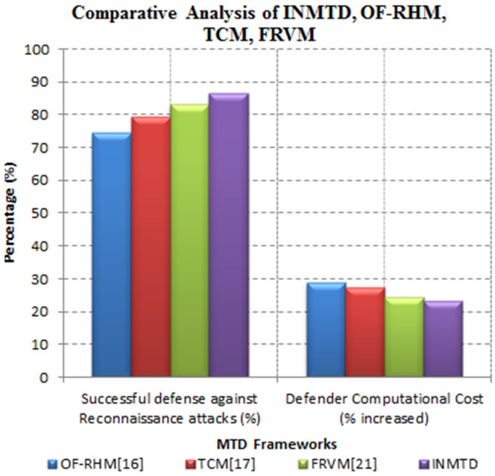

Fig. 7. Comparative analysis of INMTD with other SDN based MTD

V. CONCLUSION AND FUTURE WORK

In this paper, intent based MTD using SDN has been proposed. This is the first attempt in utilizing IBN for creating the MTD framework. The proposed INMTD model provides an efficient MTD effect at lower computational cost. INMTD successfully defended up to 86% of scanning attacks while redirecting them to shadow servers. The successful defense rate of the proposed INMTD is higher than the existing state-of-the-art SDN-based MTDs. The main computational cost is a slight increase in the number of flows while introducing the intents. This work confirms that intents are an effective mechanism for creating SDN based MTD. Regarding future work, there is a need to further investigate IBN for designing MTD with especial emphasis on enhancing Quality of Service (QoS). The current work is protecting the Data plane of SDN. In the future we plan to extend the same protocol to protect the Control plane.

ACKNOWLEDGMENT

The authors are deeply thankful to the NCBC (National Center of Big Data and Cloud Computing), NED University.

REFERENCES

[1] A. Aydeger, N. Saputro, K. Akkaya, "A moving target defense and network forensics framework for ISP networks using SDN and NFV", Future Generation Computer Systems, Vol. 94, No. 1, pp. 496-509, 2019

[2] F. Chong, R. B. Lee, C. Vishik, A. Acquisti, W. Horne, C. Palmer, A. K. Ghosh, D. Pendarakis, W. Sanders, E. Fleischman, H. Teufel, G. Tsudik, D. Dasgupta, S. Hofmeyr, L. Weinberger, National cyber leap year summit 2009: Co-chairs' report, NITRD Program, 2009

[3] J. Zheng, A. S. Namin, "A survey on the moving target defense strategies: An architectural perspective", Journal of Computer Science and Technology, Vol. 34, No. 1, pp. 207-233, 2019

[4] S. Nimmagadda, R. Kumar, P. T. Seshadri, Intent-based network security policy modification, US Patent Application Publication No. US 2019/0007453 A1, 2019

[5] C. Janz, N. Davis, D. Hood, M. Lemay, D. Lenrow, L. Fengkai, F. Schneider, J. Strassner, A. Veitch, Intent nbi–definition and principles, Open Networking Foundation, 2015

[6] B. G. Assefa, O. Ozkasap, "A survey of energy efficiency in SDN: Software-based methods and optimization models", Journal of Network and Computer Applications, Vol. 137, No. 1, pp. 127-143, 2019

[7] F. Canellas, A. Mimidis, N. Bonjorn, J. Soler, "Policy framework prototype for ONOS", IEEE Conference on Network Softwarization, Paris, France, June 24-28, 2019

[8] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, L. Benedetto, "A cyber-kill-chain based taxonomy of crypto-ransomware features", Journal of Computer Virology and Hacking Techniques, Vol. 15, No. 4, pp. 277-305, 2019

[9] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O. Connor, P. Radoslavov, W. Snow, G. Parulkar, "ONOS: Towards an open, distributed SDN OS", Hot Topics in Software Defined Networking, Chicago, USA, August 22, 2014

[10] D. Comer, A. Rastegatnia, "OSDF: An intent-based software defined network programming framework", 43rd Conference on Local Computer Networks, Chicago, USA, October 1-4, 2018

[11] R. Cohen, K. Barabash, B. Rochwerger, L. Schour, D. Crisan, R. Birke, C. Minkenberg, M. Gusat, R. Recio, V. Jain, "An intent-based approach for network virtualization", IFIP/IEEE International Symposium on Integrated Network Management, Ghent, Belgium, May 27-31, 2013

[12] G. Davoli, W. Cerroni, S. Tomovic, C. Buratti, C. Contoli, F. Callegati, "Intent-based service management for heterogeneous software defined infrastructure domains", International Journal of Network Management, Vol. 29, No. 1, pp. 46-67, 2019

[13] T. Szyrkowiec, M. Santuari, M. Chamania, D. Siracusa, A. Autenrieth, V. Lopez, J. Cho, W. Kellerer, "Automatic intent-based secure service creation through a multilayer SDN network orchestration", IEEE/OSA Journal of Optical Communications and Networking, Vol. 10, No. 4, pp. 289-297, 2018

[14] Y. Tsuzaki, Y. Okabe, "Reactive configuration updating for intent-based networking", International Conference on Information Networking, Da Nang, Vietnam, January 11-13, 2017

[15] M. Pham, D. B. Hoang, "SDN applications-the intent-based northbound interface realisation for extended applications", IEEE NetSoft Conference and Workshops, Seoul, South Korea, June 6-10, 2016

[16] J. H. Jafarian, E. A. Shaer, Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking", First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, August 13-17, 2012

[17] H. Q. Zhang, C. Lei, D. Chang, Y. J. Yang, "Network moving target defense technique based on collaborative mutation", Computers & Security, Vol. 70, No. 1, pp. 51-71, 2017

[18] A. Chowdhary, A. Alshamrani, D. Huang, H. Liang, "MTD analysis and evaluation framework in software defined network (MASON)", ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, Tempe, USA, March 19-21, 2018

[19] J. Wang, F. Xiao, J. Huang, D. Zha, H. Hu, H. Zhang, "Chaos: An SDN-based moving target defense system", Security and Communication Networks, Vol. 1, Article ID 3659167, 2017

[20] Z. Zhao, F. Liu, D. Gong, "An SDN-based fingerprint hopping method to prevent fingerprinting attacks", Security and Communication Networks, Vol. 2017, Article ID 1560594, 2017

[21] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J. H. Cho, T. J. Moore, "FRVM: Flexible random virtual IP multiplexing in software-defined networks", 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering, New York, USA, August 1-3, 2018

[22] S. Achleitner, T. F. L. Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, R. Chadha, "Deceiving network reconnaissance using SDN-based virtual topologies", IEEE Transactions on Network and Service Management, Vol. 14, No. 4, pp. 1098-1112, 2017

[23] M. H. H. Khairi, S. H. S. Ariffin, N. M. A. Latiff, A. S. Abdullah, M. K. Hassan, "A review of anomaly detection techniques and distributed denial of service (DDoS) on software defined network (SDN)", Engineering, Technology & Applied Science Research, Vol. 8, No. 2, pp. 2724-2730, 2018

[24] M. Ramzan, M. S. Farooq, A. Zamir, W. Akhtar, M. Ilyas, H. U. Khan, "An analysis of issues for adoption of cloud computing in telecom industries", Engineering, Technology & Applied Science Research, Vol. 8, No. 4, pp. 3157-3161, 2018

[25] M. Roesch, "Snort: Lightweight intrusion detection for networks", 13th Systems Administration Conference, Seattle, USA, November 7–12, 1999

[26] B. Lantz, B. Heller, N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks", 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, USA, October 20-21, 2010

[27] G. F. Lyon, Nmap network scanning: The official Nmap project guide to network discovery and security scanning, Insecure, 2009