ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse421032021.pdf
https://doi.org/10.30534/ijatcse/2021/421032021

Moving Target Defense for SDN-Based Cloud Datacenter Network Protection

[1] Tamesgen Bekele, [2] Senthil Kumar A, [3] Sisay Muleta, [4] Bekele Worku
[1] PG Scholar, School of Computing and Informatics, Dilla University, Ethiopia, bekeleteme@gmail.com
[2] Asst. Professor, School of Computing and Informatics, Dilla University, Ethiopia, asenthilkumarcse@du.edu.et
[3] PG Chair, School of Computing and Informatics, Dilla University, Ethiopia, sisaym@du.edu.et
[4] Dean, School of Computing and Informatics, Dilla University, Ethiopia, bekelew@du.edu.et

ABSTRACT

The significant advance of software Defined Networking (SDN) technology has enabled several complex system operations to be highly dynamic, flexible and robust; particularly in terms of programmability and controllability with the help of SDN controllers. Accordingly, many security operations have utilized this capability to be optimally deployed in a complex network using the SDN functionalities. Moving target defense (MTD) has emerged as an adaptive and proactive defense mechanism aiming to thwart a potential attacker. The key underlying idea of MTD is to increase uncertainty and confusion for attackers by changing attack surface (i.e., system or network configurations) that can invalidate the intelligence collected by the attackers and interrupt attack execution; ultimately leading to attack failure.
In this research, by leveraging the advanced SDN technology, the model of MTD using SDN-based system framework design is proposed. The model uses a runtime model that allows the proposed framework to infer the current state of the system. Based on the obtained information, the MTD mechanism using SDN can provide proactive, adaptive and affordable defense services for the exploitable aspects of the cloud datacenter network to increase uncertainty and complexity to the attackers and reduce the likelihood of an attack and minimize cloud security risk. The research also validates the outperformance of the proposed MTD technique in attack success rate via simulation on SDN-based cloud datacenter network experiments in a virtualized environment.

Key words: Software-Defined Network (SDN), Moving Target Defense (MTD)

1. INTRODUCTION

The newly introduced concept like multi-tenancy, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management efforts and reduced costs are among the main reasons that leveraged many enterprises, agencies and organizations to migrate their traditional datacenter to cloud [2, 3]. Cloud Computing appears as a computational paradigm as well as distribution architecture and its main objective is to provide secure, quick, convenient data storage and net computing service, with all computing resources visualized as services and delivered over the Internet. Various prominent features like scalability, flexibility, agility and reduced operational complexity through optimized and efficient computing have attracted the attention of many companies and organizations to shift from traditional data centers to cloud and rely on it to address a diverse set of user needs of access and for greater resource utilization which accommodates rapidly changing business needs [3].

In addition, the large scale of the clouds itself, the advent of mobile devices with direct access to cloud infrastructure amplify cloud vulnerabilities and threats. As a cloud computing is more extended and its utilization increases, it becomes prone to network infrastructure related potential and successful and potential attacks such as Distributed Denial of Service, hacking, stealing sensitive information, performing malicious code execution and compromising vulnerable virtual machines (VMs) which can happen in a high possibility in cloud compared to the traditional computing [7, 8].

In traditional datacenter networks, each network switch has its own control logic, which individually decides its behavior based on the information obtained from its neighbors. The traditional network approach is inefficient when it comes to the cloud data center, where a higher density of servers provides multiple VMs that dynamically transform from time to time. Moreover, there exist a series of problems, including tightly coupling of the cloud operation with the underlying forwarding infrastructure[9, 29], the timely increasing migration of individual and organization data into cloud, the possible lack of proper installations of network firewalls and the unnoticed security configurations within clouds networks, increasing the Internet dependency as a main communication medium for cloud access, and lack of coordinated and resilient defensive mechanism. These problems further lead to a rise in the volume of security problems by making the cloud much easier to be accessed and learned by adversaries' action on behalf of legitimate users as well as increase the security risk.
To address security problems, different research work and come up with a variety of different reactive defensive techniques and solutions, including newer software as well as hardware with built-in security features, higher-security network protocols, and expensive malware detection software like antivirus programs, firewalls, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), penetration testing, and so on. Despite firewall deployment, most enterprise networks have many public and private hosts accessible from outside. Using the existing dynamic IP

assignment techniques like Dynamic Host configuration Protocol (DHCP) does not protect from scanning, and using Network Address Translation (NAT) makes it difficult to reach legitimate hosts remotely[10, 11,37].

While defensive approaches have grown significantly in complexity and size over many years the adversaries still effectively learn the target and break through or bypass firewalls & IDS and easily compromise critical resources of the cloud datacenter network [12, 13]. It is because all those defense approaches were designed to detect when a problem gets to exist (i.e. reactive) or to prevent further damage once a breach has been detected. In other words, those detection-based security protection approach was tilt the balance to attackers by giving an extremely valuable and asymmetric advantage giving them the time to figure out the deployed defense mechanism, to perform reconnaissance of the target system, to study and determine potential vulnerabilities in a cloud datacenter network to choose the time to launch attack. Once attacker gains illegal accesses privilege/compromised the targeted system resource, they keep such privilege for a long period without being detected [14-16].

Moreover, since the current network configuration is static in its nature which is easily attacked and illegal access privilege is maintained for an extended period of time those security solutions aren't always enough: to effectively defend increasingly complex and intelligent penetration of datacenter network intrusion and vulnerability attacks [12]. Or to change various cloud datacenter network parameters dynamically to avoid illegal access at its initial stage by reducing a probability of an attacker's windows of successful attack [13]. For instance, the zero-day threats attacks which exploit undetected vulnerabilities in applications and may have been exploited by attackers for weeks, months, or even years.

1.1. Software Defined Networking (SDN)

Traditional datacenter network consists of hosts interconnected by forwarding devices (Figure 2. 1) that run proprietary operating systems and vendor-specific protocols which are separately configured in a tedious process in which network operators translate high-level network policies into device-specific low-level commands. They are decentralized control, complex and hard to manage and their network infrastructure does not give an efficient performance. One of the reasons is that the control and data planes are vertically integrated and vendor specific. Another, concurring reason, is that typical networking devices are also tightly tied to line products and versions. In other words, each line of product may have its own particular configuration and management interfaces, implying long cycles for producing product updates (e.g., new firmware) or upgrades (e.g., new versions of the devices)[9, 19].

Software-Defined Networking (SDN) created an opportunity for solving above long-standing problems. As a new way of network security architecture, SDN recently emerged and points to a brand-new path for building dynamic and proactive defense systems in cloud datacenter by separating the functionality of forwarding devices i.e. data planes from control planes. Basically, this decoupling enabled new network architecture: SDN that has unique capabilities such as centralized control, flow abstraction, dynamic updating of forwarding rules and software-based traffic analysis. SDN simplifies the management of complex flows, enables programmability and provides better virtualization [25, 49].

Network security is a notable part of cyber security and is gaining attention. Traditional network security practices deploy firewalls and proxy servers to protect a physical network. Due to the heterogeneity in network applications, ensuring exclusive accesses by legitimate network applications involves implementation of a network-wide policy and tedious configuration of firewalls, proxy servers, and other devices. In this aspect, SDN offers a convenient platform to centralize, merge and check policies and configurations to make sure that the implementation meets required protection thus preventing security breaches proactively. Moreover, SDN provides better ways to detect and defend attacks reactively. Ability to collect network status of SDN allows analysis of traffic patterns for potential security threats. Attacks, such as low-rate burst attacks and Distributed Denial-of-Service attacks, can be detected just by analyzing traffic patterns. At the same time, SDN provides programmatic control over traffic flows [48].

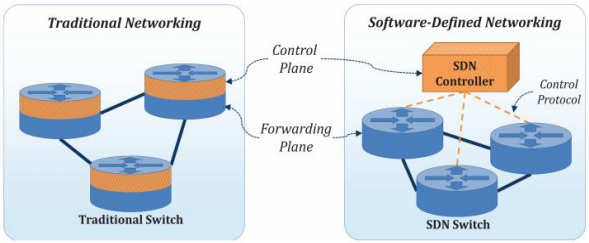

Fig. 1. SDN vs Traditional networking [49].

The need by service providers to rapidly and cost-effectively deliver network services has led to the emergency of network function virtualization (NFV), which enables network function to be deployed as a software instance. Therefore, the NFV is a network architecture paradigm that makes use of virtualization technologies to move toward a new way of designing, deploying and managing network service. Software defined networks (SDN) and network function virtualization (NFV) are innovative technologies that enable network flexibility, increase network and service agility, and support service-driven virtual networks using concepts of virtualization and softwarization. Collaboration of these two concepts enable cloud operators to offer network-as-a-service (NaaS) to multiple tenants in a data center deployment. While NFV deals with virtualization of network function, SDN introduces programmability and automation of virtual network function (VNF). SDN provides centralized view and control of the network, which can play a crucial role in achieving efficient orchestration and automation of Virtual Network Functions (VNF) [15, 49]. SDN plays a key role for network virtualization in cloud computing. Network virtualization is to segment the physical network resources in cloud data centers into smaller segmentations and lease it to cloud tenants, like leasing VMs in clouds enabled by host virtualization [29].

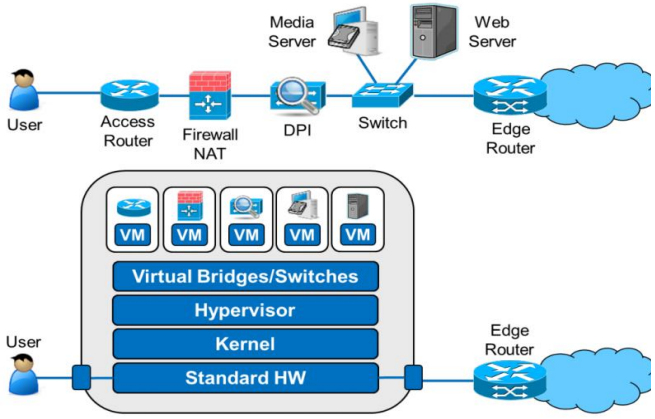

Fig. 2. Traditional vs Virtualized Approach [15].

Meanwhile as illustrated in figure 2, SDN architecture lets the underlying infrastructure to be abstracted from applications and network services in three planes,application, control and data planes. In such cases, the high-level declarative policies and dynamic updating of forwarding rules / flow rules are specified and managed by the SDN controller. The SDN controller also polls the flow statistics from network devices periodically and provides a global view of the network state in the real-time. The ability to view network state in real-time, and programmatically control network behavior by evaluating flow statistics from network elements as well as the consolidation of policies at the central SDN controller enhances consistency of dynamic and proactive datacenter network adaptation/configuration through OpenFlow protocol, helping to make attacks detection and prevention by hardening network reconnaissance. The application layer consists of customized network applications that specify data management rules and upgrade logic to the controller and the controller uses the logic from the smart application to make forwarding decisions in the data plane.

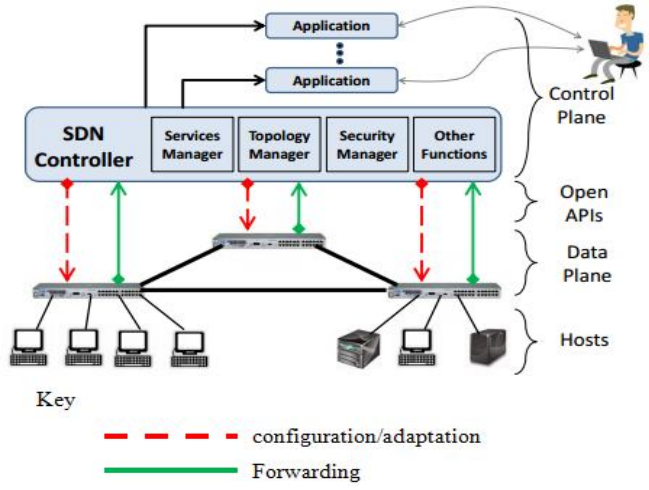

Fig. 3. SDN Architecture [9]

In such case, network control is no longer needed in each network element for policy enforcement, instead, SDN introduced a new component: the centralized SDN controller which provides better visibility and security policy enforcement capability that specify at high-level for network as whole compared to traditional networks [27]. The basic SDN architecture with a logically centralized SDN controller as illustrated in figure 1 contained information about the entire network to command and control all devices in the cloud datacenter network centrally through consolidated policy in the central controller. The SDN controller also can assess reachability information of all the hosts in a network through standard and open interface, such as the OpenFlow protocol.

1.2. System Architecture and Framework

Dynamic MTD Mechanism using SDN-Based System The control plane (i.e. SDN controller) does the abstraction of the current state of the network configuration with the functional requirements periodically from Infrastructural Logic Model, which is the runtime system model of the physical or virtual network to build the dynamic and proactive adaptations. The general operation of the framework which runs on the SDN controller is determined by Adaptation Engine, which injects proactive adaptations to the current configuration overtime intervals based on the current state of the network configuration from Infrastructural Logic Model. The proactive adaptations are carried out by Configuration Manager, which creates a set of adaptation rules / set of configuration policies that are implemented on the underlying network via secured OpenFlow protocol. The overall abstraction of the current configuration of the physical or virtual network and the creation of the configuration policies are managed by SDN controller that provides a global view of the network state in real-time by polling the flow statics from network forwarding devices (i.e. SDN switch) periodically under data plane.

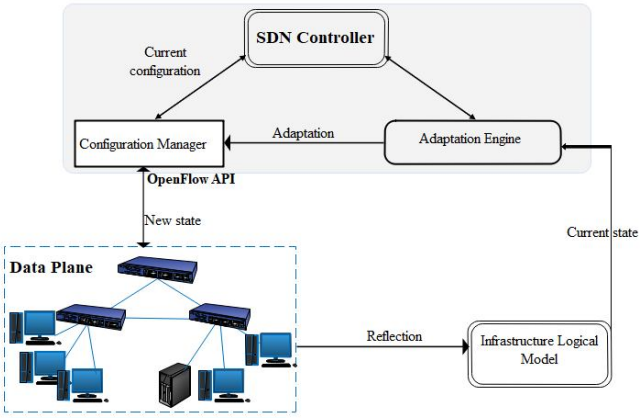

Fig. 4. Dynamic MTD mechanism using SDN-Based framework

The Dynamic MTD using SDN framework makes proactive adaptations of the exploitable aspects of the cloud datacenter network overtime by Adaptation Engine and Configuration Manager.

The SDN controller which does the overall flow abstractions and traffic analysis. Then the adaptation engine determines an appropriate set of adaptation rules or a set of configuration policies (e.g. adapting / mutating address of VMs or application running on the VMs, etc.) based on the current state configuration of the network. Finally, the adaptation engine selects the configuration rule and sends it to the configuration manager in order to implement the adaptation proactively over an extended interval on the data plane network infrastructure through Secured OpenFlow protocols.

Proactively changing of the vulnerable aspects of the network configuration overtime interval will affect the attack success probability of an attacker in two ways: Enforcing an attacker to spend more time and effort to make network reconnaissance that enables them to identify topological information that will be useful for further attack by constantly changing the current configuration network/attack surfaces overtime. An attacker can't keep his/her access opportunity for a long period of time by timely/randomly adapting or reconfiguring the datacenter network resources.

However, due to the dynamic and on-demand access nature of cloud computing this can also imply two basic challenges for the MTD using SDN system. One, while a network configuration can be made more dynamic through proactive adaptation or mutation overtime, the MTD mechanism must ensure the way of allocating required virtual network services to the legitimate users in the middle of the proactive adaptation intervals. Nevertheless, once the target system is compromised, an attacker would regain his/her ability by having access privilege and locating the resources, limiting the effectiveness of the proposed system. To address this problem, the MTD mechanism makes dynamic and proactive adaptation randomly overtime interval to decrease the damage incurred through compromised systems by reducing the overall attack surface which limits attacker knowledge of the locations information of other virtualized resources. The second challenge is that, while the MTD mechanism can proactively adapt applications or resources, the transition process will disrupt services and introduce a necessary overhead cost. Thus, the effect of adaptations on both the system performance and security improvement must be understood so that the appropriate compromise can be made.

Therefore, to implement the ground-truth of high-level Dynamic MTD using SDN-based system architecture, this research derives from the proposed Proactive Network Adaptation System (PNAS) system model.

1.3. The Proactive Network Adaptation System (PNAS) System Model

The need by service providers to rapidly and cost-effectively deliver network infrastructure services has led to the emergence of Software Defined Networking (SDN), which enables network functions to be deployed as software instances [15]. SDN promises network function virtualization, scalability, flexibility and agility for service providers offering cloud-based network connectivity services to multiple tenants. Since Virtualized Network Functions (VNFs) for different tenants (clients) in a multi-tenant virtualized network environment often share the same physical infrastructure, appropriate security mechanism need to be implemented to protect tenants from each other's malicious actions, which may be intentional or unintentional as well as internal or external. This research addresses the problem of static and reactive defense nature of traditional datacenter networks to a virtualized network cloud datacenter environment. These solutions need to be defined and implemented within well-defined tenant network functions virtualization framework. This section, therefore, presents an elaborate PNAS system model which is derived from dynamic MTD using SDN-based framework, which uses SDN for automated control and efficient network policy implementation. While PNAS implements shuffle-based proactive MTD technique deals with virtualization of network functions, SDN introduces programmability and automation of virtual network functions. The key feature of SDN is the decoupling of network control from forwarding planes [9]. SDN also provides a centralized view and control of the network, which can play a crucial role in achieving efficient orchestration and automation of the virtual network services. The proposed PNAS illustrates and evaluates how MTD and SDN can be architecturally combined, thus converging to deliver a true 'softwarized' network services environment.

The proposed Proactive Network Adaptation System (PNAS) model is derived from MTD using SDN-based cloud datacenter system framework which provide network adaptation or mutation solution for the SDN enabled cloud data center, the component and configuration policies and a set of proactive adaptation are implemented as SDN application within a centralized control entity called SDN-Controller. With the MTD mechanism, the attack surface is hardened through random adaptive approach. This approach attempts to make dynamic and proactive changes to the attack surface at runtime [11]. Adaptations by the MTD using SDN-based cloud datacenter network can be proactively overtime interval by transforming attack surface of virtual network functions (i.e. VM's). However, the problem with a "moving" system after a random proactive adaptation is how it can locate and transparently communicate virtual network services in the system with other services they depend upon for functionality.

Therefore, the Proactive Network Adaptation System (PNAS) which is implemented as a system control plane requires and knows the location information of all other components with which this component functionally depends upon in system reconfigurations. And also, the SDN controller which reduces the complexity can provide better visibility and security policy enforcement allows Configuration Manager to communicate securely with PNAS via OpenFlow API to enable reconfiguration/adaptations.

All communications between mission-critical services are controlled by PNAS through SDN controller, so even the location information of the services changed by proactive adaptations the communication can be maintained within specified adaptation time interval. In other words, the PNAS add network dynamics that prevent the attackers which attempt reconnaissance scan to access and exploit services by following a predefined path and simplify attack detection and prevention and also impose dynamics for an attacker to force them to repeatedly conduct extensive reconnaissance in re-identify service locations to increase an attack effort and cost to the attackers.

The virtual network functions (VNFs) are in fact virtual machines (VMs) instances deployed in a virtualized cloud environment. To accomplish efficient proactive adaptation, it is critical to ensure that there is no interference between the different hosts. To this end, this system model comprehensively demonstrates in the next section how the tenant sharing the same cloud environment can be effectively isolated, adapted/reconfigured and secure from each other.

The proposed Proactive Network Adaptation System (PNAS) MTD security mechanism is a set of regulatory and addressing policies that cloud operators to their data center can define to ward off internal or external reconnaissance scanning action that can compromise host security.

The PNAS system model presented framework can be viewed as tiered model consisting of the data plane (i.e. OVS connects

different VMs containing services) and the control plane (SDN controller, Configuration Manager and Proactive Adaptation) respectively. The SDN controller has complete visibility of the hosts, thus making it an appropriate point to initiate and propagate network policy rules.

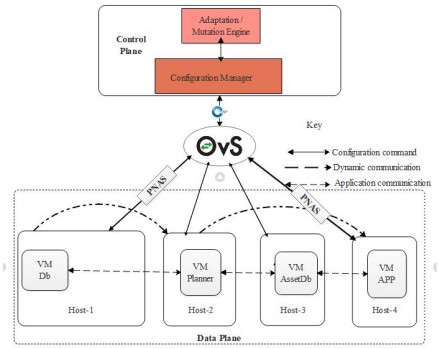

Fig. 5. Proposed Proactive Network Adaptation System (PNAS) Model

The openvSwitch and hosts shown in fig. 5 form the data plane of the proposed MTD using SDN for cloud data center network security solution. The data plane consists of virtual components including a hypervisor which is a software layer that enables simultaneous execution of multiple network operating systems (OS) in one compute node [55]. Each Host's or VM's appears as a self-contained logical machine with independent processor, memory, network interface and other computing resources having executing their required service. The hypervisor is responsible for allocating physical server's/host' resources amongst the different logical machines and ensuring that they do not disrupt each other. These logical machines are often termed virtual machines (VMs). The physical infrastructure that runs the VMs does not necessarily have to be purpose-built. Figure 4.3 illustrates the MTD components of the data plane architecture providing VMs that can run software instances implementing network functions. Typically, the physical infrastructure belongs to the cloud-based network connectivity services provider. The cloud provider [2] leases out the VMs to multiple tenants in datacenter to run their virtual network functions.

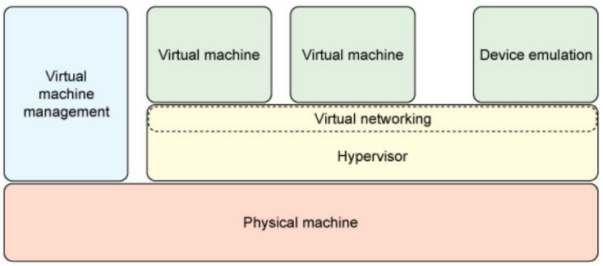

Fig. 6. Data plane components of PNAS

To optimize network communication among hosts, the edge switch or Open vswitch (OVS) can be introduced. For the framework presented in this chapter OpenFlow enabled switches: Open vSwitches (OVS) are used to facilitate communication between Hosts or VMs. Figure 6 below presents an OVS and its interfaces. The virtual interfaces (vIFs) associated with VMs communicated through the OVS to the physical interfaces (pIFs). The OVSs contain flow tables which define forwarding actions ordered by SDN controller between hosts in one OVS using vIFsand between adjacent OVSs through the pIFs. The flow table is updated by the controller through the southbound interfaces.

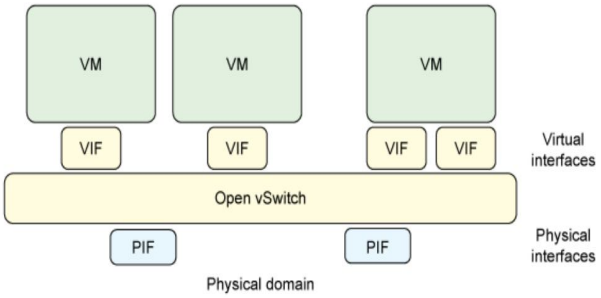

Fig. 7. Open vSwitch with virtual and physical interfaces

In the PNAS framework, adaptation engine is the main decision-making MTD component inside the control plane (SDN controller) that controls the modifiable aspects of the hosts or VMs in the datacenter network. To validate the technical merit or to predict the effectiveness of the proposed PNAS model in cloud datacenter network defense, it needs to determine the frequency of the proactive network adaptations and which aspects of the systems can be modified using the adaptation engine.

The proactive adaptation enforces configuration manager in SDN controller to manage the edge switches in the network to control packet forwarding decisions while SDN-enabled openvswitch (OVS) deal with forwarding packets. The OVS is configured to encapsulate packets that have no exact matching flow rules in flow tables, and the encapsulated packets, called "OFPT PACKET IN" packets in OpenFlow (OF) protocol, are forwarded to the SDN controller for the handling of the flow. Finally, the SDN controlled configuration manager updates the network component (physical or virtual) with current flow (configuration). This configuration is given to the SDN controlled Configuration Manager, which makes changes and provides appropriate knowledge to the PNAS component on the host. In all the cases, the changes are determined by the Adaptation Engine and sent to the Configuration Manager who makes the appropriate change in the physical or virtual system. In this research, it is assumed that the SDN controller and secure control channel are trusted; the case of the SDN controller or the control plane of the SDN being compromised by the attacker, using redundant and distributed controller MTD solution is to be set which is out of the scope of our research.

1.4. The Proactive Adaptation / Mutation

As depicted in Figure 4 of the SDN enabled cloud datacenter network, there is a series of operation or action called Adaptation that mutate the current modifiable configuration state of the data center network to a valid configuration state overtime to prevent the attack effort of the attackers by reducing the attack surface available for exploitation. This approach is also known as MTD using SDN-based approach (or PNAS). In this research, proactive adaptation is the basis for defining and evaluating the effectiveness of theMTD technique in SDN-based systems.

Therefore, PNAS system is an approach that can adapt network configuration during executions to achieve the overall security defense goal of the cloud datacenter network. To handle this specific type of transformation operation, it needs to define as:

An adaptation / mutation is a sequence of actions A = {a1, a2... ak} that transform the current configuration state, S, to a valid configuration state, Sv.

The MTD using SDN-based system Σ - is a tuple {Cs, G, P} where Cs is a configurable system, G is the set of goals which includes both service / operational goals and security goals and P is the set of configuration policies.

A critical part of a cloud computing attack is the reconnaissance scanning or exploration of the target system's configuration information. The assumption made is that if an attacker gains access i.e. breaks the reconnaissance or scanning phase of the target system to the host running multiple VMs, the attacker can then access the resources which are shared by other VMs. The attackers can be directly or indirectly as well as internally or externally connected to the SDN-enabled cloud data center network. They can run different networking probing attacks against the different hosts connected at the edge switch of the data plane to gather hosts information. For this research, the attacker's targets are the running virtual network functions running on VM's. As the first step of a cyber-kill chain, the attacker will attempt to make a reconnaissance attack. Each unique IP address is considered as an attacker.

In the proposed PNAS model, an adaptation or mutation which is a shuffle-based solution for virtualized deployment in a cloud data center network that dynamically changes IP addresses of the host. This can be deployed by a cloud operator to minimize internal attacks initiated from other tenants or from external attacks. This solution decreases the chance of an attacker using scanning tools to correctly identify reachability of potential targets. The adaptation engine inside the SDN controller periodically changes the virtual IP addresses (vIPs) in networks in intervals of adaptation time (Tm). From each tenant network mask the unused IP addresses will be randomly assigned as vIPs to the virtual network functions. The controller also maps and make attachment of real IP with vIFs of the hosts (or VMs) randomly amongst OVSs in the infrastructural logical node.

An attacker scanning the network will get different IP addresses and location overtime interval for a single Host (VM) in different time intervals. The controller is responsible for updating the flow tables in the OVSs. It maps the original configuration action rules i.e. rIP of each host to the changing vIP to enable transparent end-to-end communication. Within each proactive adaptation overtime, the host's doesn't participate in transformation. The controller will also be used to update DNS responses by responding with the vIP for the queried device. The proactive IP adaptation solution allows only the rightful owners of the virtual networks to use the original IP addresses i.e. real IP (rIP) to access the virtual network functions. The detail of the proposed PNAS algorithm is shown below:

PNAS Algorithm

```
determine unused IPs in network block
determine number of OVSs in Infrastructural model
for (packet p from OVS)
        if (p.type = Type-A DNS response for host hi) then
                set DNS address to current vIP(hi), TTL ≈ 0
        else if (p.type= TCP-SYN or UDP from hi to hj )
                if (p.src in internal)
                   install in flow in src OVS with action srcIP(p)= vIP(hi)
                 else
                   install out flow in src OVS with action dstIP(p) = vIP(hj)
        else if (p.dst = rIP) then
                if (hi is authorised tenant)
                        install in and out flows in OVS
                else (p.dst=vIP)
                install in flow in src OVS with action srcIP(p)= vIP(hi)
                install out flow in src OVS with action dstIP(p) = vIP(hj)
for all adaptation of each host hi do
        if (vIP not used)
                 set vIP(hi) to new vIP
        else
                find another vIP
                set vIP(hi) to new vIP
```

First, a tenant is authenticated and then allowed to access the services using the original flow entries created when the physical or virtual networks were deployed that define traffic flows using the rIPs. The controller acts as the central authority managing IP adaptation amongst OVSs, installing new flows and deleting old flows in the OVS and responding to DNS requests.

Figure 4.5provides a flow diagram of how a host sends network traffic to a VM on the same or different subnet in a PNAS. First, Host-1 wants to send a message to the vIP address of Host-2. Additional network management must occur to inform clients of vIPs. One potential way to share this information with the clients is through DNS cache updates that occur in sync with adaptations. Second, the SDN switch receives the request and discovers that there is no rule in the flow table to handle this type of traffic. Third, the switch asks the controller how to handle this request. The controller determines that traffic coming from VM-1's rIP (10.0.0.5) destined for VM-2's vIP (10.0.0.42) must have the source/destination headers modified. Host A's rIP must be replaced with its vIP (10.0.0.80). The destination of VM-2's vIP then translates to its rIP (10.0.0.10). After this calculation, the fourth step is to update the flow table entries on the SDN switch. Fifth, the switch conducts address translation between the rIPs and vIPs for VM-1 and 2. Finally, the switch sends the message to VM-2. Note how the SDN switch sends a request to the SDN controller when predefined behaviors are not present at the switch. The network policies set by the Configuration Manager are instantiated as rules at the controller and transmitted across the Southbound API to the data plane.

The controller has a record of all functions in each service and corresponding virtual interfaces connecting to the functions. The network is created in mininet [54] emulator which is used as an infrastructural logical model has a defined network subnet-mask that determines the number of addresses and block size that can be allocated to unique functions. The number of OVSs in the cloud environment are also noted and all usable vIF are seen as one pool of interfaces. From the unused IP address range in the vIP assigner randomly selects using random number generator and assigns vIPs for all interfaces in a VM and maps the vIFs to a different interface from the vIF pool in the cloud environment. This action is performed periodically in intervals of the selected adaptation rate (Tm). The assigner has to ensure that a vIP or vIF is not assigned to more than one function at a time to avoid collision and that it is not assigned one function in consecutive mutation intervals to increase destruction and confusion for an attacker attempting to scan the network properties.

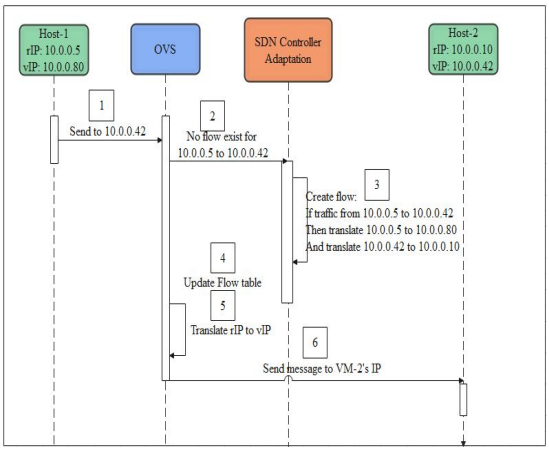

Fig. 8. PNAS Workflow diagram

When a DNS query is sent to resolve the name of a host, the DNS response is updated by the controller to replace the rIP of the network nodes with currently active vIP. The controller also sets the time-to-live (TTL) value in the DNS response to a small value. The source host can then initiate the connection using the vIP of the destination. After Tm time adaptation interval all vIP will be reconfigured meaning that successive DNS queries of the same nodes is likely to give a different vIP causing confusion and complexity for a potential attacker.
After every adaptation interval performed by the vIP assigner the configuration manager inside the controller updates the Network Address Translation (NAT) with new vIPs to corresponding static rIPs. When a source sends packets for the first time the OVS encapsulates and sends the initial packet to the controller. The controller installs flow entries in all the OVSs in the route to the destination host that translates the vIP to the rIP and initial vIF to current vIF hence packets will be forwarded using action associated with the rIP and original vIF to ensure transparent communication. All the OVSs in the route will be configured to route traffic based on vIP addresses. Successive packets can now be matched and forwarded by the OVS within the virtual networks according to the installed flows. The NAT application guarantees transparent end-to-end reachability of hosts, because the real IP (rIP) to virtual IP (vIP) and virtual Interface (vIF) translation for a specific connection remains unchanged regardless of subsequent adaptation. The algorithm for the proposed system is presented below. The SDN controller first identifies the unused IP addresses in the network and adds vIFs from all OVSs into a pool of vIFs in the cloud environment. This IP range will be used to assign the vIP for the VNFs. The controller also hops vIFs of the services to randomly selected interfaces from the pool. The controller is also responsible for responding to DNS requests by giving out the current vIP of the functions.

Therefore, to evaluate the effects of the mechanism, it should be able to characterize the exploration surface / the attack surface that the attacker must explore before attacking. Such an exploration surface is represented in the form of PNAS as illustrated in figure 4.3.1 due to continuous proactive changes overtime in system configurations to predict the frequency and effectiveness of MTD mechanism can affect the on-going attack effort. Using this PNAS, simulation-based experiments were conducted to show at what point (time) should the MTD using SDN-based mechanism make an adaptation / reconfiguration to increase the effectiveness of the MTD system, while maintaining a reasonable cost. More details of the implementation of simulation-based experiments were presented in the next chapter.

2. SIMULATION SETUP AND RESULTS

2.1. PNAS Scenario

To run a PNAS in Software Defined Network-based cloud data center network, an effective controller that handles the traffic efficiently were needed. So the PNAS model programs the network infrastructure i.e. data planes using RYU controllers. This section presents the experimental scenarios carried out on a RYU controller and network constructed in an Mininet environment and controlled remotely by the custom adaptation engine and configuration Manager which injects timely random proactive residing within the PNAS controller. The OpenFlow protocol is a Communication protocol used to communicate between controller and networking devices which includes network switches and routers. OpenFlow protocol is considered as a facilitator of Software Defined Networking as it is vendor independent. In this PNAS implementation have used OF version 1.3 to create and handle secure and effective communication between Data place and control plane. 1000 experiments were conducted each for four different adaptation intervals (30, 60, 120 and no adaptation, static) using the attack interval-time between adjacent attacks of 30s. The adaptation interval implies the time interval between the controller's next mutations. The table below shows the experimental parameters used to validate the PNAS model.

Table 2. Simulation Parameters

Network size	10-to-10
Adaptation Technique	IP Address and vIF Mutation
Adaptation Interval(s)	30, 60,120 and static
Number of Controller	1 RYU Controller
Open vSwitchs (OVSs)	3 OVSs with 8 host
Number of experiments	100-1000
Attack interval	30s
Attack type	Reconnaissance or Scanning attack

2.2. Experimental Setup and Evaluation

For the realistic evaluation of the PNAS system model, the controller was deployed onto a virtualized network, constructed with Mininet. The security and performance of the PNAS controllers was measured on the virtualized network. Mininet provides the ability to create large virtualized networks that are portable, do not require expensive hardware, and are easily shareable for others to confirm results. Off the shelf, Mininet provides pre-made network elements such as hosts, OpenFlow switches, and SDN controllers. All network elements run a Linux sub-system that operates real-world protocols and applications. Alternatives to Mininet included an abstracted network platform that simulated relevant network mechanisms, or a physical SDN-based network. The abstracted network platform would not have captured how PNAS affects existing real-world network infrastructure, therefore, avoiding a realistic implementation of PNAS. Whereas, PNAS was not

deployed on a physical SDN-based network due to time and lack of resources constraints related to the configuration of physical network elements. Mininet was preferred over both options due to the ease of configuration and the use of real-world applications, protocols, and off the shelf network elements.

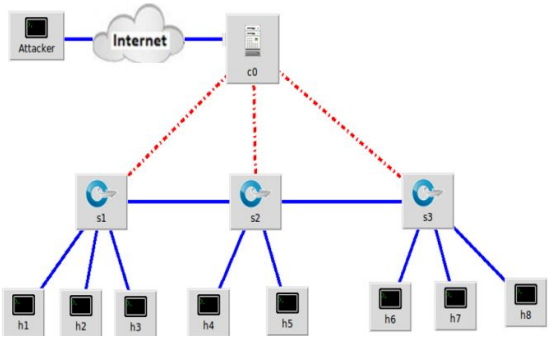

Fig. 8. Proposed Network Topology

Mininet creates the network using command line interface (CLI) while the graphical user interface (GUI) is generated in MiniEdit. The CLI for creating the network is given below in Figure 9.

```
root@mtdlab:~# sudo mn --custom ~/mininet/custom/topoForPNAS.py --topo mytopo --
controller remote --switch ovsk
*** Adding switches
*** Adding hosts
*** Adding links
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h2) (s1, h3) (s1, s2) (s2, h4) (s2, h5) (s2, s3) (s3, h6) (s3, h7)
 (s3, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

Fig. 9. Proposed Network Topology of the virtualized network

Where, the parameters mn start the CLI Mininet, --custom is used to start saved topology --toporuns a topology containing 3 switch and 8 host (server), --switch ovsk uses OpenvSwitch, --controller remoteuse external OpenFlow controller. All the nodes have assigned a unique IP address and MAC address. The IP address and MAC address for node h1, h2, h3, h4, h5, h6, h7 and h8having an IP address of '10.0.0.1', '10.0.0.2', '10.0.0.3', '10.0.0.4', '10.0.0.5','10.0.0.6','10.0.0.7' and'10.0.0.8' respectively with default mac address. After creating virtual SDN network topology, different xterm started to open on the hosts. Each xterm corresponds to nodes 1 to 8, the switch and the controller.

To implement the RYU controller of PNAS component, the directory of RYU had accessed and executed the Proactive network adaptation application which supported OpenFlow 1.3 in the xterm window titled as "controller c0 (root)". The file located in the folder/RYU/app named MTDSDN.py, which require some time to connect to OpenvSwitch (OVS).

```
root@mtdlab:~/ryu# ryu-manager --verbose ryu/app/MTDSDN.py
loading app ryu/app/MTDSDN.py
loading app ryu.controller.ofp_handler
instantiating app ryu/app/MTDSDN.py of MovingTargetDefense
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK MovingTargetDefense
  PROVIDES EventMessage TO {'MovingTargetDefense': set([])}
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPPacketIn
  CONSUMES EventMessage
BRICK ofp_event
  PROVIDES EventOFPSwitchFeatures TO {'MovingTargetDefense': set(['config'])}
  PROVIDES EventOFPPacketIn TO {'MovingTargetDefense': set(['main'])}
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPPortDescStatsReply
  CONSUMES EventOFPEchoReply
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPHello
  CONSUMES EventOFPErrorMsg
  CONSUMES EventOFPEchoRequest
Creating Event
EVENT MovingTargetDefense->MovingTargetDefense EventMessage
('Random Number:', 18)
********** {'10.0.0.8': '10.0.0.27', '10.0.0.5': '10.0.0.28', '10.0.0.4': '10.0.0.29', '10.0.0.7': '
10.0.0.30', '10.0.0.6': '10.0.0.31', '10.0.0.1': '10.0.0.32', '10.0.0.3': '10.0.0.33', '10.0.0.2': '
10.0.0.34'} **********
********** {'10.0.0.27': '10.0.0.8', '10.0.0.28': '10.0.0.5', '10.0.0.29': '10.0.0.4', '10.0.0.33':
'10.0.0.3', '10.0.0.32': '10.0.0.1', '10.0.0.31': '10.0.0.6', '10.0.0.30': '10.0.0.7', '10.0.0.34':
'10.0.0.2'} **********
```

Figure 10. RYU Controller running PNAS

After that RYU controller able to activate on remote IP address (127.0.0.1) to host machine Mininet, status of a RYU controller connecting OpenFlow enabled switch of PNAS application shown in Figure 10. In each proactive adaptation, random number is generated in every 30 second. After the study of the experiment different data analyzed, which shows the expected results.

2.3 Proactive Network Adaptation Results

From the implemented network design to evaluate the PNAS through the RYU controller, the primary metrics like deterrence, proactive Adaptation rate and flow-table size between proactive adaptations has assessed under reachability of ICMP traffic using iperf3 and ping benchmarking tools.The results below presents the experimental data collected using the Wireshark tool [56] based on the experimental parameters. For different adaptation intervals a study was conducted on how many scans are needed and time for successful identification of services running on specified port. A study for flow table length was made by varying the number of sessions established per second for different adaptation / mutation intervals.

The proposed PNAS solution for MTD using SDN-based cloud datacenter network deployment reduces the chances of reconnaissance in identifying IP addresses of the VMs hosting tenant services and their location within the SDN-based cloud. The results below demonstrate the effectiveness of this Moving Target Defense (MTD) mechanism.

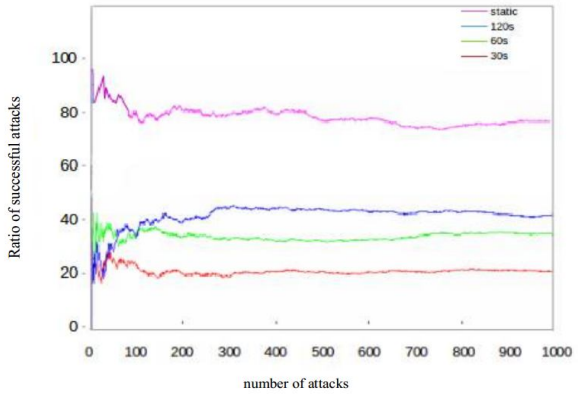

Fig. 10. Success rate of individual attacks

Figure 10 shows the success rate of each individual attack between nodes for different adaptation intervals. For all the adaptation intervals, the success rate fluctuates for the initially small number of samples and then becomes more stable as the number of attacks increases. The figure shows that as the adaptation interval is reduced (adaptation frequency increased), the individual attack success ratio also decreases, and as can be observed from the figure success rate is reduced by 40% for the adaptation interval of 30s compared to a static network. This implies that as the adaptation frequency increases an attacker needs to perform more scans than in the static network. It can therefore be concluded that the adaptation ratios increase as the mutation frequency increases.

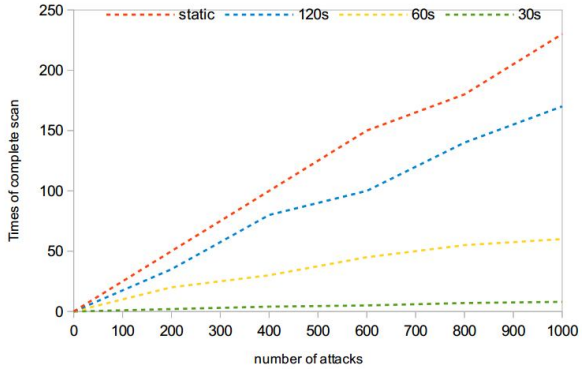

Fig. 11. Complete attack against target PNAS system model

Figure 11 also clearly shows the effect of the proposed MTD mechanism. When the adaptation interval is reduced, the success rate of correctly identifying the VMs decreases. This experimentation was conducted for 3000 seconds. When the configuration is static, the number of completed attacks is 240 out of 1000, while for the mutation interval of 120s the number is reduced to 50 and an adaptation interval of 30s allows only 5 successful attacks. This figure also explicitly shows that as the mutation frequency increases more time and number of attacks are required to successfully identify virtual network services therefore it can also be concluded that the adaptation rate and deception ratios increase as the frequency increases. To achieve better deception and adaptation ratios higher adaptation frequency should be used.

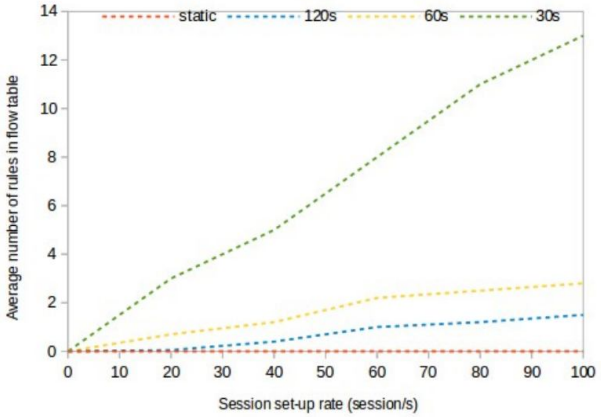

Fig. 12. Length of flow-table per adaptation interval

However, this will require a network with large block size and unused addresses. A network with the largest unused address space can be assigned a non-repetitive vIP address in adjacent adaptation intervals causing more confusion to an attacker. The size of the network block therefore directly affects the effectiveness of the PNAS solution. Although it has been observed that increase in the frequency of adaptation increases complexity and difficulty for attackers, this on the other hand increases the length of flow tables in the OVS switches and also causes some jitter. Figure 12 illustrates that an increase in the number of sessions per second results in a longer flow table with increase of mutation frequency.

3. CONCLUSION

Experimental set-ups were implemented using the tools proposed for the PNAS security solutions using SDN cloud datacenter deployment were evaluated and analyzed. The PNAS concept is analyzed and evaluated based on three metrics: deception, adaptation rate and flow table length against scanning tools that are normally used by attackers to reconnaissance network and its vulnerabilities. An analysis made shows that increase of mutation frequency increases the deception and adaptation or mutation ratio hence making it harder for an attacker to correctly hit on target virtual network functions. The results also implies that an increase in IP address block enables non-repetitive use of a virtual IP address (vIP) and also brings more confusion and deception to an attacker. However, it has also been illustrated that shorter adaptation or mutation intervals results in larger flow table size in the OVSs. This research designs and evaluates security solution framework that illustrate the benefits of integrating MTD mechanism in SDN controller architecture to simplify implementation and management of security in the dynamic MTD using SDN environment. These MTD using SDN based solutions allow automation of security policies hence enabling Unpredictability of network, scalability, programmability, and network and service agility at the same time reducing CAPEX. The controller can be used to modify or upgrade security rules based on newly discovered threats requirements. The experimental results demonstrate, the attack success likelihood reduced as increasing the frequency or rate of the random adaptations configurations of the vulnerable attack surface overtime that increases the uncertainty and complexity to the potential attackers.

Generally, the simulation results based on the design of the framework confirm the defense approach that proactively adapting and configuring the exploitable aspects of the cloud datacenter network randomly overtime can decrease the attack success probability of the attacker by creating a completely chaotic system environment. The results showed, the attack success probability is reduced as increasing frequency or rate of adaptations / mutations by MTD using SDN.

REFERENCES

1. Alhebaishi N, Wang L, Jajodia S, et al. Threat Modeling for Cloud Data Center Infrastructures, 2016[C]. Springer, 2016. 302-319.
2. Xu X. From cloud computing to cloud manufacturing [J]. ROBOT CIM-INT MANUF, 2012, 28(1):75-86.
3. Hashizume K, Rosado D G, Fernández-Medina E, et al. An analysis of security issues for cloud computing [J].

Journal of Internet Services and Applications, 2013, 4(1):5.
4. Rekha P M, Dakshayini M. Dynamic network configuration and Virtual management protocol for open switch in cloud environment, 2015[C]. IEEE, 2015. 143-148.
5. Chung C. SDN-based Proactive Defense Mechanism in a Cloud System [M]. Arizona State University, 2015.
6. Hong J B, Kim D S. Assessing the effectiveness of moving target defenses using security models [J]. IEEE T DEPEND SECURE, 2016,13 (2):163-177.
7. Xiong Z. An SDN-based IPS Development Framework in Cloud Networking Environment [Z]. Arizona State University, 2014.
8. R. Hwang, H. Tseng and Y. Tang, "Design of SDN-Enabled Cloud Data Center," 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, 2015, pp. 950-957, doi: 10.1109/SmartCity.2015.193.
9. Ali S T, Sivaraman V, Radford A, et al. A survey of securing networks using software defined networking [J]. IEEE T RELIAB, 2015, 64(3):1086-1097.
10. Wang W. A Cyber-security Defense Method Using Docker Containers [Z]. Vanderbilt University, 2015.
11. DeLoach S A, Ou X, Zhuang R, et al. Model-driven, moving-target defense for enterprise network security [M]//Springer, 2014:137-161.
12. Zhuang, R., DeLoach, S. A., & Ou, X. (2014). A model for analyzing the effect of moving target defenses on enterprise networks. Proceedings of the 9th Annual Cyber and Information Security Research Conference- CISR '14. doi:10.1145/2602087.2602088.
13. Okhravi H, Shrobe H. Moving Target Techniques: Cyber Resilience throught Randomization, Diversity, and Dynamism [Z]. MASSACHUSETTS INST OF TECH LEXINGTON LEXINGTON United States, 2017.
14. Zhuang R, DeLoach S A, Ou X. Towards a theory of moving target defense, 2014[C]. ACM, 2014. 31-40.
15. Yackoski J, Li J, DeLoach S A, et al. Mission-oriented moving target defense based on cryptographically strong network dynamics, 2013[C]. ACM, 2013. 57.
16. Open Networking Foundation, "OpenFlow Switch Specification" https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf specifications/openflow/openflow-spec-v1.4.0.pdf, 2013.
17. McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69-74.
18. E. Skoudis, Counter Hack Reloaded, Second Edition: A Step-by-step Guide to Computer Attacks and Eective Defenses, Second. Upper Saddle River, NJ, USA: Prentice Hall Press, 2005, isbn: 9780131481046.
19. Shi Y, Zhang H, Wang J, et al. CHAOS: An SDN-Based Moving Target Defense System [J]. SECUR COMMUN NETW, 2017, 2017.
20. B. Pfaff et al., "The design and implementation of openvswitch," 12th USENIX symposium on networked systems design and implementation (NSDI 15), pp. 117-130,2015..
21. NITRD C. IWG: Cybersecurity game-change research and development recommendations [Z]. 2013.
22. Casola V, De Benedictis A, Albanese M. A moving target defense approach for protecting resource-constrained distributed devices, 2013[C]. IEEE, 2013. 22-29.
23. Wang L, Wu D. Moving target defense against network reconnaissance with software defined networking, 2016[C]. Springer, 2016. 203-217.
24. Peng W, Li F, Huang C, et al. A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces, 2014[C]. IEEE, 2014. 804-809.
25. S. Azodolmolky, P. Wieder and R. Yahyapour, "SDN-based cloud computing networking," 2013 15th International Conference on Transparent Optical Networks (ICTON), Cartagena, 2013, pp. 1-4, doi: 10.1109/ICTON.2013.6602678.
26. Sonchack J, Smith J M, Aviv A J, et al. Enabling Practical Software-defined Networking Security Applications with OFX., 2016[C].2016. 1-15.
27. Chowdhary A, Pisharody S, Huang D. SDN based Scalable MTD solution in Cloud Network, 2016[C]. ACM, 2016. 27-36.
28. Moody W C, Hu H, Apon A. Defensive maneuver cyber platform modeling with Stochastic Petri Nets, 2014[C]. IEEE, 2014. 531-538.
29. Jungmin Son and RajkumarBuyya. 2018. A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing. ACM Comput. Surv.51, 3, Article 59 (May 2018).https://doi.org/10.1145/3190617.
30. Evans D, Nguyen-Tuong A, Knight J. Effectiveness of moving target defenses [M]//Springer, 2011:29-48.
31. L. L. Zulu, K. A. Ogudo and P. O. Umenne, "Emulating Software Defined Network Using Mininet and OpenDaylight Controller Hosted on Amazon Web Services Cloud Platform to Demonstrate a Realistic Programmable Network," 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), PlaineMagnien, 2018, pp. 1-7, doi: 10.1109/ICONIC.2018.8601254.
32. Zhuang R, Zhang S, DeLoach S A, et al. Simulation-based approaches to studying effectiveness of moving-target network defense, 2012[C].2012. 1-12.
33. Sheldon F T, Vishik C. Moving toward trustworthy systems: R&D Essentials[J]. COMPUTER, 2010, 43(9):31-40.
34. Douglas C. MacFarland and Craig A. Shue. "The SDN Shuffle: Creating a Moving-Target Defense Using Host-based Software-Defined Networking". In: Proceedings of the Second ACM Workshop on Moving Target Defense. MTD '15. Denver, Colorado, USA: ACM, 2015, pp. 37–41. ISBN: 978-1-4503-3823-3.
35. JafarHaadiJafarian, Ehab Al-Shaer, and Qi Duan. "An effective address mutation approach for disrupting reconnaissance attacks". In:IEEE Transactions on Information Forensics and Security 10.12 (2015), pp. 2562–2577
36. https://www.programmersought.com/article/62304394471/.(Accessed on March, 2020).
37. Al-Shaer E, Duan Q, Jafarian J H. Random host mutation for moving target defense, 2012[C]. Springer, 2012. 310-327.
38. Yackoski J, Bullen H, Yu X, et al. Applying self-shielding dynamics to the network architecture[M]//Springer, 2013:97-115.
39. Nagineni, &Satheesh. Performance analysis and evaluation of software defined networking distributed

controllers in datacenter networks. International Journal of Computational Systems Engineering. 5. 61. 10.1504/IJCSYSE.2019.10019690, 2019.

40. Al-Shaer E. Toward network configuration randomization for moving target defense[M]//Springer, 2011:153-159.
41. Kampanakis P, Perros H, Beyene T. SDN-based solutions for Moving Target Defense network protection, 2014[C]. IEEE, 2014. 1-6.
42. Barik M S, Sengupta A, Mazumdar C. Attack graph generation and analysis techniques [J]. DEFENCE SCI J, 2016,66(6):559.
43. Sandra Scott-Hayward, Design and deployment of secure, robust, and resilient SDN Controllers, 978-1-4799-7899-1/15/$31.00 [c] 2015 IEEE
44. Iman E., Ankur C., Dijiang H., Software Defined stochastic model for moving target defense, international Afro-European conference for Industrial Advancement, 188-197,2016.
45. M. Dunlop, S. Groat, R. Marchany, and J. Tront, “Implementing an IPv6 Moving Target Defense on a Live Network,” in National Symposium on Moving Target Research, June 2012.
46. Cataldo et al., "A novel approach for integrating security policy enforcement with dynamic network virtualization," Network Softwarization (NetSoft), 2015 1st IEEE Conference, 2015.
47. Open Network Foundation, “Software-Defined Networking: The New Norm for Networks”, ONF White Research, 2012
48. Wenfeng X, Yonggang W,Chuan H, et al. “Survey on Software Defined Networking”, IEEE communication surveys & tutorials, VOL. 17, NO. 1, FIRST QUARTER 2015.
49. Gui-linCai, Bao-sheng Wang, Wei Hu, and Tian-zuo Wang. “Moving target defense: state of the art and characteristics”. In: Frontiers of Information Technology & Electronic Engineering 17.11 (Nov. 2016)
50. M. Carvalho and R. Ford, "Moving-Target Defenses for Computer Networks," in IEEE Security & Privacy, vol. 12, no. 2, pp. 73-76, Mar.-Apr. 2014, doi: 10.1109/MSP.2014.30.