

Review Article

Moving Target Defense-Based Denial-of-Service Mitigation in Cloud Environments: A Survey

Minh Nguyen and Saptarshi Debroy 

City University of New York, New York, USA

Correspondence should be addressed to Saptarshi Debroy; saptarshi.debroy@hunter.cuny.edu

Received 23 September 2021; Revised 28 January 2022; Accepted 19 February 2022; Published 22 March 2022

Academic Editor: AnMin Fu

Copyright © 2022 Minh Nguyen and Saptarshi Debroy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increased frequency and intensity of denial-of-service (DoS) attacks on critical cloud-hosted services, resource adaptation schemes adopted by the cloud service providers (CSPs) need to be intelligent. Specifically, they need to be adaptable to attack behavior and be dynamic to curb resource over-utilization. The concept of moving target defense (MTD) has recently emerged as an effective and agile defense mechanism against DoS attacks that particularly target cloud-hosted applications. However, the existing surveys that seek to explore this space either focus more on MTD for generic cyberattack mitigation or on DoS attack defense on cloud systems. In this survey, we particularly provide an in-depth analysis on how MTD can help recover critical cloud assets in the face of DoS attacks and how emerging programmable technologies such as software-defined networking (SDN) can be leveraged to achieve that goal. Unlike existing surveys, we categorize DoS attacks on cloud platforms based on their working mechanism. We also discuss the non-MTD-based DoS defense strategies for both cloud and non-cloud infrastructures in order to highlight the pros and cons of MTD-based strategies. We introduce MTD working mechanisms and present how existing research is envisioning MTD's application in mitigating DoS attacks, both with and without SDN. We also take an in-depth look at the testbed implementations and resilience and performance evaluations of MTD approaches. Finally, we articulate the existing challenges in MTD for DoS mitigation in cloud systems and how these challenges are shaping the future research in this domain.

1. Introduction and Background

1.1. Cloud vs. Classical Computing. With the high demand of online services that are spatially and temporally diversified, data migration to cloud platforms has proliferated due to its cloud resources' scalability and elasticity [1]. Before the cloud era, most of the enterprise assets in terms of services and data were stored in dedicated physical hardware—the bigger the enterprise assets, the greater the need for such physical resources. However, in this solution, resources do not scale well with increased load and consequently cost explodes with increased resources. Cloud computing on the other hand provides on-demand cyber resources (i.e., computing, storage, and networking) over the Internet with subscription-based pay-as-you-go pricing model for its customers [1]. This enables enterprises that are consumer service or content providers to rent elastic cyber resources

from public or private cloud service providers (CSPs), such as Amazon Web Services (AWS) [2], Microsoft Azure [3], Google Cloud [4], GENI [5], and CloudLab [6] instead of buying, owning, and maintaining physical data centers and servers. Consequently, cloud has become part of the critical infrastructures for hosting essential services and data in areas such as finance, education, government services, and healthcare.

As illustrated in Figure 1, simple abstraction of cloud infrastructure usually consists of four layers: (1) the bottom-most hardware layer provides the physical resources such as CPUs, storage units, and network resources (switches and cables), (2) the virtualization layer in the middle hosts the hypervisor (a.k.a virtual machine monitor or VMM) that creates the virtualized environment, (3) the middleware layer cross-cuts other layers by providing distributed services, such as resource management and monitoring, and (4)

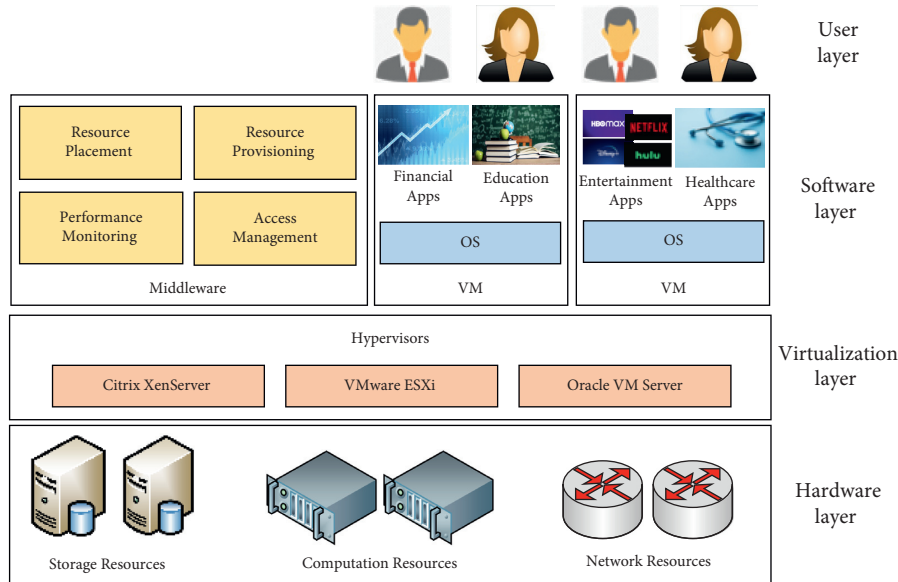


FIGURE 1: Layered architecture of a cloud environment hosting critical services through provisioning distributed and elastic services.

the software layer runs the virtual machines (VMs) that host critical services/applications. The service model offered by the CSPs to the client services can also come in different packages: (i) infrastructure-as-a-service (IaaS) provides a data center and a way to host client VMs and data, (ii) platform-as-a-service (PaaS) provides a programming environment to build and manage the deployment of an application, and (iii) software-as-a-service (SaaS) provides delivery of software to the service users. Seamless integration among different layers of the cloud architecture and easy implementation of different service models within the same datacenter are made possible by adoption of programmable technologies, such as software-defined networking (SDN) [7], OpenFlow [8, 9], OpenDaylight [10], and OpenStack [11].

1.2. Adoption of SDN for Efficient Cloud Management. It can be argued that if “virtualization” was the most critical technology towards realization of cloud computing, then “softwarization” is the primary reason behind the “cloudification” of most of the critical services. Also, no technology other than SDN [7, 8, 12] was more responsible behind such “softwarization.” SDN allows the networks to be programmable, making them more flexible and agile, i.e., network changes are made through software rather than hardware. To achieve this, SDN decouples the control plane (i.e., network control) and data plane (i.e., network functions) and enables the control plane to become directly programmable. This allows the underlying network infrastructure to be abstracted for applications and network services (in the app plane) [7] as shown in Figure 2. The control plane hosts a centralized SDN controller that acts as the brain of the network and manages flow control to data plane via southbound APIs (e.g., OpenFlow [8, 9]). Data plane consists of the network devices (e.g., switches/routers) that follow the rules handed down from the controller and

perform the corresponding forwarding functions. The interactions between the app plane and the controller use northbound APIs (e.g., RESTful [13]). Unlike traditional networks where each network device contains the entire network stack of data plane, control plane, and app plane (as shown in Figure 2), SDN with its decoupled and centralized control is more dynamic, adaptable, and agile.

The recent advances in software-defined technologies that use OpenFlow protocol have made it easier to manage and control distributed cloud data centers across geographic boundaries. SDN has allowed public and private CSPs to implement fine-grained and dynamic network control (i.e., routing, switching, identity and access management, and resource provisioning) across its data centers based on diverse vectors such as data type and size, sources and destinations of data streams, privacy and security requirements, and resource availability to name a few. The role and impact of SDN towards efficient management of distributed cloud services cannot be overstated, none more so than for institutional private clouds that support data-intensive science. Figure 3 shows an exemplar institutional private cloud infrastructure that features SDN with OpenFlow switches at strategic traffic aggregation points within the campus and backbone networks that feature science DMZ (demilitarized zone or perimeter network, sits between an internal network and an external network [14]) for friction-free data-intensive science workflows [15]. SDN provides centralized control on dynamic science workflows over a distributed network architecture and thus allows proactive/reactive provisioning and traffic engineering of flows in a unified, vendor-independent manner [9]. It also enables fine-grained control of network traffic depending on the QoS requirements of the application workflows. In addition, OpenFlow-enabled switches help in dynamic modification of security policies for large flows between trusted sites when helping them dynamically bypass the campus firewall [16]. The figure also shows the infrastructural components of the institutional

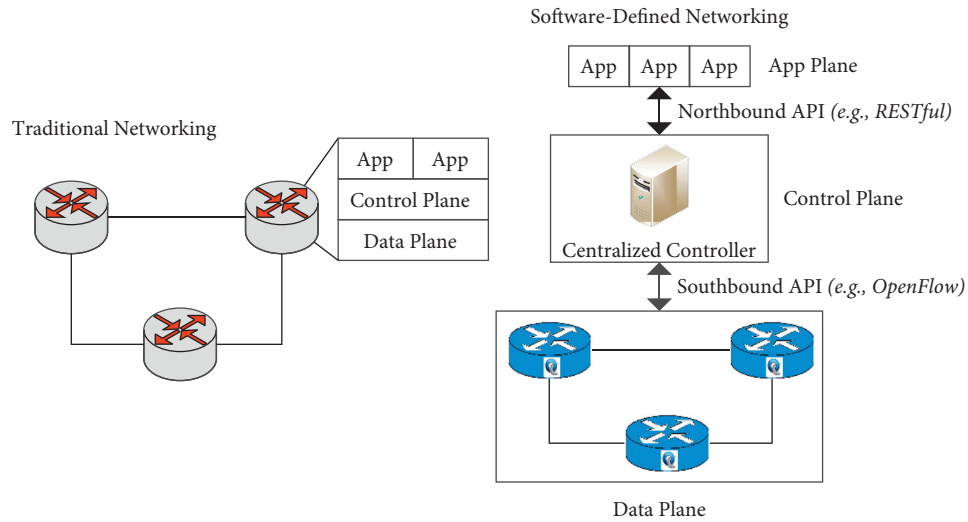


FIGURE 2: Traditional networking vs. software-defined networking.

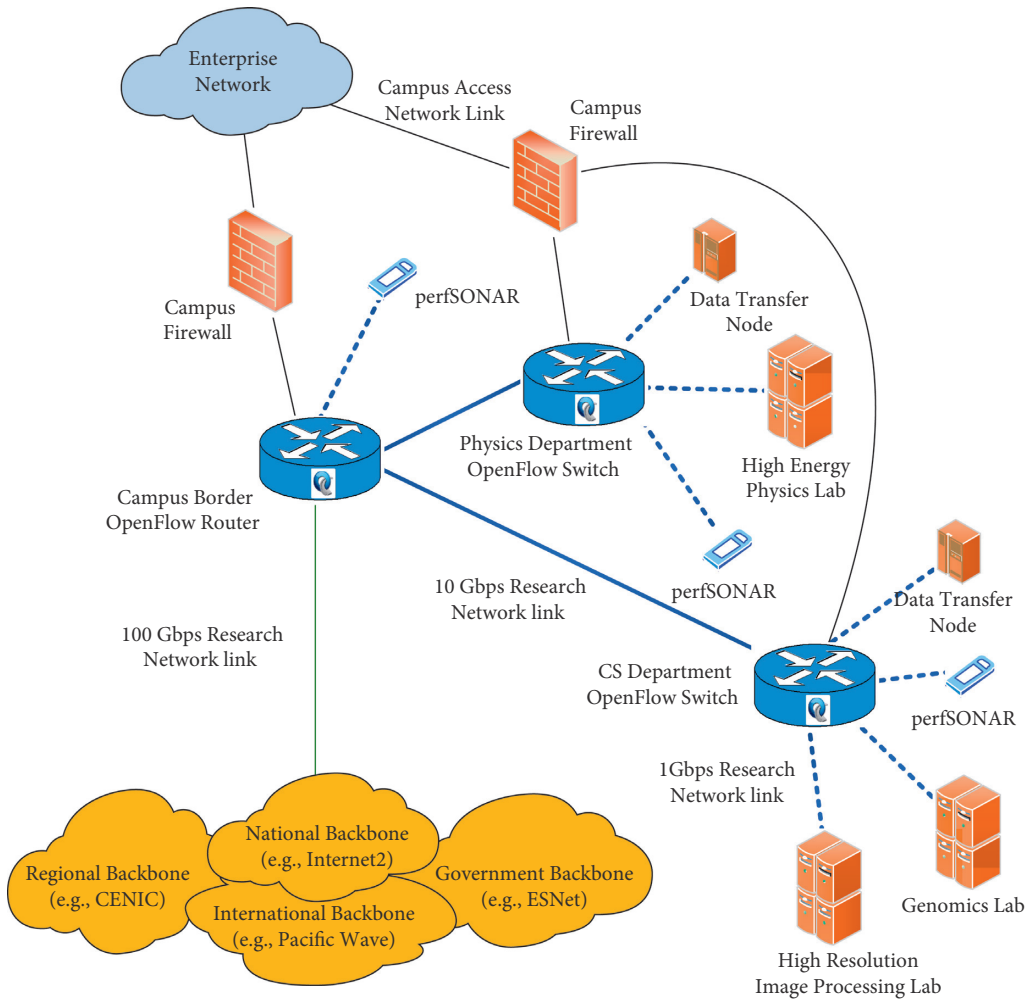


FIGURE 3: An exemplar institutional private cloud with science DMZ infrastructure enabled by SDN and OpenFlow.

science DMZ within the campus network. Normal application traffic traverses paths with intermediate campus firewalls and reaches remote collaborator sites or public

cloud sites over enterprise IP network to access common web applications. However, data-intensive science application flows from research labs that are “accelerated” within

science DMZs bypass the firewall to the high-speed backbones.

1.3. Denial of Service on Cloud and Mitigation Challenges. The proliferation of cloud-hosted enterprise and scientific service has made the entire cloud ecosystem an enticing target of cyberattacks, in particular to denial-of-service (DoS) attacks that lead to loss of availability (LoA) of services through resource exhaustion. A DoS attacker typically accomplishes such exhaustion by flooding the target directly or indirectly with malicious traffic (usually with spoofed source IP addresses) till the target's resources are overwhelmed when it cannot respond or simply crashes [17], thus starving the legitimate users (of the cloud services) from critical services. A distributed denial-of-service (DDoS) attack is an extreme version of DoS attack. Although conceptually DDoS attack mechanism is the same as DoS, in DDoS, the attacker commands and controls an army of bots or zombies (botnets, usually contain malware-infected smartphones, personal computers, IoT devices, routers, etc.) [17, 18] that collaboratively and simultaneously bombards the target with attack intensity clocking hundreds of Gbps and Tbps. Such volume can cause even the most resilient of the systems to buckle effectively, quickly resulting in service unavailability to a large population of users. Besides, when a sudden and large surge in traffic happens at a server, it is called a flash crowd or flash event [19]. Although flash events happen with far less frequency than DoS attacks and are usually caused by legitimate traffic, they still pose issues for CSPs as their characteristics are very similar to those of DoS/DDoS attacks. In fact, some attackers try to masquerade their DoS attacks as flash events [20].

In Akamai Technologies' 2018 State of the Internet report [21], overall DoS/DDoS attacks went up (16%) in Q4 2018 in comparison to 2017 showing a steady year-to-year increase. The same report from 2019 [22] shows that DoS/DDoS attacks continue to target the cloud industry as more than 80% of the attack events have targeted cloud-based consumer applications (e.g., gaming or Internet and telecom) as shown in Figure 4. The largest DDoS attack ever recorded happened in February 2020 when AWS cloud services saw peak attack traffic at a rate of 2.3 Tbps [23, 24]. High-volume DoS attacks are not only restricted to consumer cloud applications; collaborative cloud environments such as GitHub [25] are also targets, e.g., 2018 DDoS attack on GitHub had a volume of 1.35 Tbps via 126.9 million packets per second (pps) [26]. Lack of adequate defense and recovery strategies to counter against such attacks can impact cloud service provider (CSP) reputation and cause millions of dollars in damages to cloud tenants.

The DoS attack defense challenges within a cloud platform are more severe in the following two ways. Firstly, a cloud environment becomes a vulnerability amplifier to traditional cyber security threats due to the fully distributed and highly elastic nature of the infrastructure resources designed to serve a large population. For example, one of the largest DDoS attacks in the history was launched on cloud-based DNS servers of Dyn, Inc. [27] in 2016 (peak at 1.2

Tbps) [28] that percolated to different layers of the Internet, crippling not only the CSPs such as AWS but also popular cloud-hosted content providers such as Netflix and Twitter. The attack impacted millions of users on the East Coast of the United States of America with close to 12 hours of service outage, resulting in loss of money and subscription [28].

Secondly, new means of attack exist that specifically target the vulnerable areas cloud environments such as application multitenancy, decentralized network management, and third-party broker services (between the CSP and the consumers). For example, when hackers carry out a cyberattack on a cloud-based service, they can either try launching the attack from outside targeting the server IP address and/or the DNS or they can infiltrate the internal network of the CSP hosting streaming services and target vulnerable virtual machines (VMs) that have security soft spots, catering to a large population of consumers for greater impact. Although such network infiltration-based attacks are difficult to carry out requiring increased attack budget, most sophisticated attacks on cloud-based services are network infiltration based. What makes matters worse is that such infiltration-based attacks endanger the entire cloud environment, i.e., individual VMs, underlying operating systems, and hardware infrastructure by making them vulnerable to a plethora of other attacks [29].

1.4. Moving Target Defense (MTD). In order to tackle the aforementioned challenges, the cloud security community and even federal organizations are exploring "Cyber Agility and Defensive Maneuver (CAADM)" mechanisms [30] that are (a) agile in response to attack detection, (b) cost-effective for the CSP, and (c) sophisticated in tackling intelligent attack strategies. The goal is for such mechanisms to allow real-time service restoration through agile cloud resource adaptations once a DoS attack is detected. The same mechanisms can also limit proliferation of detected attacks within the cloud infrastructure through preventive VM resource maneuvers. Among the CAADM mechanisms, moving target defense (MTD)-based resource obfuscation/adaptation strategies are most effective to protect critical cloud-hosted applications [31]. For instance, MTD-based mechanisms are used to perform both (i) *proactive* resource adaptation, to detect a DoS attack and act defensively before major damage is inflicted, and (ii) *reactive* resource adaptation, to act defensively after an attack has occurred. At the same time, MTD-based mechanisms are amenable to leverage the emerging software-defined networking (SDN) [7, 8, 12] paradigm to achieve dynamic network resource management [32].

However, there are three distinct issues that makes the design of such MTD-based CAADM strategies non-trivial. Firstly, with every dynamic resource adaptation, the CSP encounters cost involving wastage of cloud network/compute/storage resources, which becomes especially prohibitive for proactive adaptations. However, the alternate approach of infrequent adaptations can leave the application vulnerable to DoS threats. Thus, there is a need to optimize the frequency of proactive adaptations. Secondly, with either

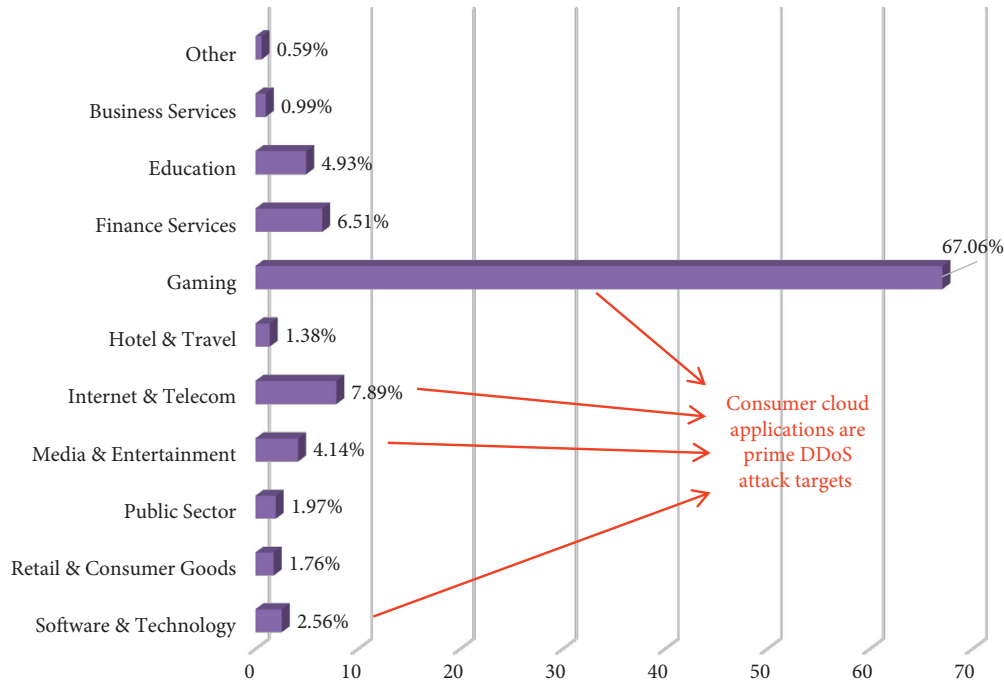


FIGURE 4: Akamai Technologies' 2019 survey showing DoS attack frequency by industry with more than 80% of attacks targeting consumer cloud applications.

proactive or reactive resource adaptation, the legitimate users of a cloud-hosted application will experience service interruptions and quality of experience (QoE) degradation to some extent. Such degradation can be sustained if the resource adaptations are suboptimal and do not capitalize on the inherent heterogeneity of CSP resources to optimize performance. Thus, there is a need to optimize the CSP resource utility in the adaptations without noticeably impacting the end-user performance. Thirdly, successful MTD-based defense implementations need to possess the potential for deception, wherein a quarantine environment traps the attacker without his/her knowledge to learn more about the attack strategy, while the defense adaptations are progressing to continue service to legitimate users.

1.5. Contribution of This Survey. These challenges have prompted cyber security community across academia, federal government, and private enterprise to explore the utility of MTD-inspired attack reflection and recovery strategies in cloud environments that are efficient and yet cost-effective. In recent years, researchers have also conducted extensive surveys on a decade worth of works in this space primarily targeting two focus areas: (i) MTD-inspired cyber defense mechanisms [33–36] and (ii) DoS/DDoS defense mechanism for cloud-hosted services [37–39]. Compared to these works, our survey seeks to focus more on the application of MTD strategies in clouds infrastructure against DoS attacks and how programmable technologies such as SDN are being leveraged to implement such strategies.

In particular, the main contributions of this survey are as follows:

- (i) Unlike existing surveys, we categorize DoS attacks on cloud platforms based on their working mechanism, e.g., volume-based, protocol-based, and application-based attacks in order to shed more light on attack process.
- (ii) Unlike other MTD focused surveys, we discuss the non-MTD-based DoS defense strategies for both cloud and non-cloud infrastructures. This gives us a better perspective on the pros and cons of MTD-based strategies against DoS and DDoS.
- (iii) In this survey, we introduce MTD's working mechanism and present how existing research envisions MTD's application in mitigating DoS attacks, both with and without SDN. We take a unique approach to categorize MTD-based strategies on the basis of maneuvering techniques, e.g., IP shuffling, live migration, and proxy management. Unlike existing surveys, this approach of categorization highlights the different MTD strategies at various network abstractions.
- (iv) We take an in-depth look at the testbed implementations and resilience performance evaluations of MTD approaches. For example, we discuss how existing research uses cloud testbeds, hardware testbeds, simulation, and sometimes combination of these to demonstrate strategy effectiveness. We also showcase different performance (i.e., usability) and security metrics used for such demonstration.
- (v) Finally, we articulate the existing challenges in MTD for DoS mitigation in cloud systems and how these challenges are shaping future research in this domain.

The rest of the paper is organized as follows. Section 2 discusses the comparison of our survey with existing surveys. Section 3 discusses DoS/DDoS attack classification. Section 4 introduces different MTD-based mitigation strategies. Section 5 discusses the evaluation methods and metrics. Section 6 highlights the existing challenges and future directions. Section 7 concludes the paper. The overall paper organization is illustrated in Figure 5.

2. Comparison with Existing Surveys

In Table 1, we compare the focus of our survey against recent popular surveys that primarily focus on MTD-based cyberattack mitigation. In one of the early ones [33], Cai et al. conducted a thorough survey on MTD-based mitigation techniques. The authors presented a function-and-movement model to provide different perspectives for understanding MTD research works. With this model, they systematically surveyed MTD works based on three main areas, e.g., theory, strategy, and evaluation. Within these areas, the authors further classified MTD into subcategories based on techniques and characteristics of MTD strategies. However, the survey needs a better categorization of MTD techniques and implementations keeping the most recent work in mind. It also misses some key perspectives such as types of DoS attacks and state-of-the-art experimental testbeds for evaluation of MTD-based strategies.

In 2019, Zheng and Namin published an extensive survey on MTD-based cyber defense mechanisms [34]. This work is focused on architectural aspects and classifications of MTD strategies. The authors categorized MTD strategies based on the level of implementation within the system stack, e.g., OS level, software/application level, and network level. For each level, the authors further categorized MTD based on techniques such as IP randomization, virtualization, and decoy among others. However, this survey is not focused on cloud systems and SDN capabilities. Furthermore, the survey lacks a comprehensive discussion on the evaluation methods and metrics for the existing MTD techniques.

Recently, in [35], Sengupta et al. presented an extensive survey on MTD techniques for advanced persistent threat (APT) [40, 41] in SDN-based cloud environments. In this survey, the authors provided an in-depth analysis on the implementation and evaluation of MTD techniques and how technologies such as SDN and network function virtualization (NFV) can aid MTD implementation. The authors categorized MTD techniques based on the interrelationship between different phases of APT. Moreover, the survey introduces a common terminology library that can help readers understand more about the underlying assumptions and threat models of existing MTD techniques. Besides, the authors conducted a thorough study on MTD evaluation techniques with various security and usability metrics. However, the survey lacks DoS focus and does not provide an extensive study on non-MTD-based DoS mitigation techniques that are essential to appreciate and understand the pros and cons of MTD.

In another recent work [36], Cho et al. conducted a comprehensive survey on MTD's application for a wide range of cyberattacks in cloud, SDN, and IoT environments. They classified MTD techniques with their respective pros and cons based on three types of operations: shuffling, diversity, and redundancy. The different types of MTD techniques are discussed in the context of different attack vectors, e.g., shuffling, diversity, and redundancy. Besides, the authors also extensively discussed the evaluation methods and metrics (performance and security) used to validate the performance of the MTD techniques. Although similar to [35], this work is very comprehensive, and it does not focus on DoS or non-MTD-based works targeting DoS attacks.

Table 2 illustrates the comparison between our survey and other surveys that focus on DoS and DDoS-based attacks and defenses in cloud environments. In [37], Yan et al. performed extensive survey about DoS/DDoS attacks in cloud infrastructures and especially in SDN environments. The authors studied how DoS attacks can be launched in cloud environments and how defense mechanisms can be designed against those attacks by exploiting SDN programmability. However, the authors did not discuss the current research on state-of-the-art experiments and evaluation methods. Recently, in [38], Agrawal and Tapaswi also conducted a comprehensive survey about DoS in cloud environments. The authors classified DoS attacks based on various forms of high-rate and of low-rate attacks and discussed their strategies and impacts. Besides, they categorized the defense approaches and their performances based on multiple evaluation metrics; however, they only partially analyzed attacks in SDN environments. More recently, Yurekten and Demirci conducted a thorough survey on SDN-based defense for cyberattacks that includes DoS/DDoS attacks [39]. The authors categorized cyberattacks by examining the five-phase cyber threat intelligence. However, this survey does not focus on cloud environments. As for defense, they provided SDN-based defense mechanisms that can be used to cope with aforementioned cyber threats based on detection, prevention, and mitigation aspects. Finally, the authors discussed the evaluation techniques based on addressed threat category, defense type, defense strategy, and underlying solution approach.

Compared to these surveys, our survey focuses more on cloud infrastructure and SDN aspects of the network design. We provide in-depth analysis of how MTD strategies are implemented on SDN environments to mitigate DoS attacks and how those approaches stack up against each other, both in terms of usability and security. In this survey, we specifically classify DoS attacks into three categories based on the mechanism, i.e., volume-based, protocol-based, and application-based. For the MTD strategies, we categorize them based on the maneuvering techniques such as IP shuffling, proxy, and live migration. In our survey, we also present non-MTD approaches for defending against DoS in cloud environments. We also discuss the existing challenges in MTD-based DoS defense and future directions to address those challenges.

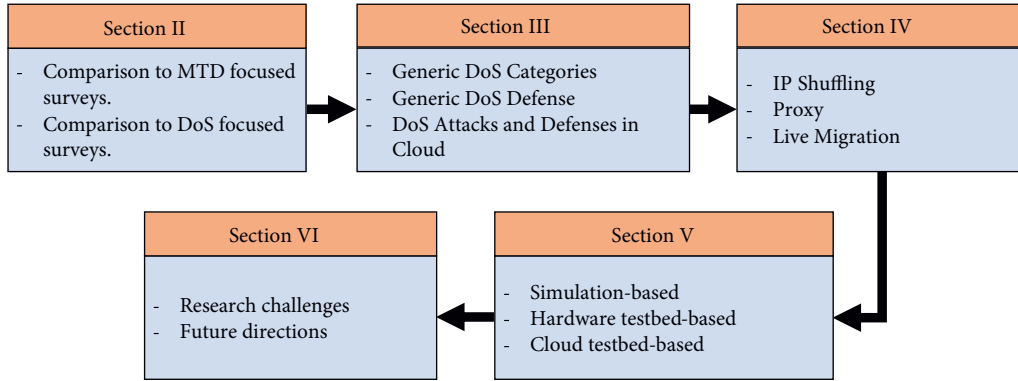


FIGURE 5: Overall survey roadmap and contributions.

TABLE 1: Comparison of key contributions of MTD.

Related work	Cai et al. (2016) [33]	Zheng and Namin (2019) [34]	Sengupta et al. (2020) [35]	Cho et al. (2020) [36]	Our survey
Focus on cloud and SDN	Partially	Partially	Comprehensively	Comprehensively	Comprehensively
In-depth analysis On DoS attacks	No	No	No	No	Comprehensively
Non-MTD Defense methods	No	No	No	Partially	Comprehensively
MTD techniques and implementations	Partially	Comprehensively	Comprehensively	Comprehensively	Comprehensively
Evaluation methods	Comprehensively	Partially	Comprehensively	Comprehensively	Comprehensively
Future directions and challenges	Partially	Partially	Comprehensively	Comprehensively	Comprehensively

TABLE 2: Comparison of existing surveys about DoS/DDoS and cyberattacks.

Related work	Yan et al. (2016) [37]	Agrawal and Tapaswi (2019) [38]	Yurekten and Demirci (2020) [39]	Our survey
DoS in cloud	Comprehensively	Comprehensively	Partially	Comprehensively
DoS in SDN	Comprehensively	Partially	Comprehensively	Comprehensively
Evaluation methods	No	Comprehensively	Comprehensively	Comprehensively
Focus on MTD	No	No	No	Comprehensively

3. DoS Attacks and Mitigation in Cloud

3.1. Overview of DoS Attacks. Broadly, DoS attacks can be categorized into three different types: volumetric attacks, protocol attacks, and application attacks [42]. In real-world scenarios, attacks could be launched as a combination of the three in order to increase the devastating effects. In this section, we discuss these attack strategies, whereas the mitigation strategies will be discussed further in Section 3.2.

3.1.1. Volumetric Attacks. Volumetric attacks are classic DoS attack where the goal is to deny service by typically creating congestion and saturation of bandwidth at the target (e.g., server) and the target network. This makes it impossible for legitimate users of the service to communicate with the server under attack. Typical examples of volume-based attacks are UDP flood, ICMP flood (a.k.a. ping flood), and amplification attacks (a.k.a. reflection attacks). In UDP flood, a large volume of UDP packets

bombards a server that makes the server check for processes that are listening to the ports and respond to each UDP packet. This leads to denial of service for the regular clients. UDP flood as a matter of fact is behind the very first documented DDoS, the attack on University of Minnesota in July, 1999 [43–45]. Ping flood (ICMP flood) is another type of volume-based attack where the objective is to consume the victim server’s bandwidth usually by sending ICMP echo requests as fast as possible. Due to the way ICMP works (for each request, there is a reply) [46], ping flood ends up consuming the attacker’s bandwidth as well. However, there are ways to work around this feature.

A more sophisticated and potentially dangerous type of volumetric attacks are amplification attacks (a.k.a. reflection attacks) where instead of the real target, a vector is targeted that can reflect and amplify the attack traffic towards the real target. Typical example is the DNS amplification attack where the attacker makes large number of requests to DNS (Domain Name System) [47] servers with spoofed source IP addresses and the destination is changed to the target’s IP

address. As a result, the DNS servers forward the volume of responses to the victim. Other popular amplification attacks include NTP (Network Time Protocol) [48] and SSDP (Simple Service Discovery Protocol) [49] amplification. In these, the attackers typically exploit the bad design of a UDP-based request and response protocol (e.g., DNS or NTP) and trigger a significantly larger number of responses than the original amount. Although the recent discoveries of new amplification vectors are rare, once such attacks hit the target, more often than not they lead to devastating consequences. For example, the previously mentioned recent DoS attacks on AWS [24] and GitHub [26] were both amplification attacks that exploited rather newly found vulnerabilities of CLDAP (Connectionless Lightweight Directory Access Protocol) [50] and Memcached [51] protocols.

3.1.2. Protocol Attacks. Protocol attacks exploit weaknesses in the working mechanisms of network and transport layer protocols to cause a service denial by over-consuming the server resources and other equipment in the network infrastructure (e.g., firewalls and load balancers). Typical examples of protocol attacks include SYN flood and IP fragmentation attacks. One of the first recorded DoS events on the Internet in 1996 was a SYN flood attack [52] followed by many other high-profile DoS attacks in the history [52–54]. Typically TCP (Transmission Control Protocol) [55] uses a three-way handshake to establish a connection: (1) the client sends a SYN message to the server to request a connection; (2) the server acknowledges the request by sending SYN + ACK message back to the client and leave an open port waiting for the final acknowledgment; and (3) the client responds with the final ACK message and the connection is established. In SYN flood, the attacker exploits the last feature by not only not sending back the final ACK packet but also sending more SYN packets leading to denial of service at the server for other legitimate users due to the lack of ports.

Another example of protocol attacks is IP fragmentation attack which exploits the network maximum transmission unit (MTU) [56]. IP fragmentation process mandates that any transmitted IP packets larger than the network MTU (e.g., 1500 bytes for Ethernet [57]) will be broken into IP fragments which will later be reassembled at the final destination [58]. The attacker exploits this mechanism by preventing the packets to reassemble at destination (e.g., by only sending a part of the packet), resulting in service unavailability. Other protocol attacks include ping of death and Smurf that exploit ICMP [59]. However, they are largely considered solved for contemporary hardware/software systems [59].

3.1.3. Application Attacks. The final broad category refers to application attacks where the attackers seek to exhaust the target server's resources by exploiting the vulnerabilities of network applications (e.g., web servers). In general, application attacks usually are considered most sophisticated and mitigation techniques are rather complex. Typical examples

of application attacks include HTTP flood and low-and-slow attacks. In HTTP flood, the attacker floods the target web server with HTTP GET packets (used to request for images, files, etc. from a server) and/or HTTP POST packets (used to send data to a server and/or database in order to create/update a resource) [60]. This consumes not only bandwidth but also disk space and available memory of the target server. Besides the previously mentioned attack in 2018 [26], GitHub had also suffered the largest DDoS attack ever in 2015 which was a HTTP flood attack [23]. Other popular application attacks are low-and-slow attacks. Low-and-slow attacks operate by requesting the targeted server to execute some tasks and then sending data to the server at a very slow pace in order to keep the tasks unfinished for long time. As a consequence, the server has to always keep the connection open in order to finish the requested tasks which in turn denies other tasks from other legitimate users. Examples include attacks using tools such as Slowloris [61, 62] or "R.U.D.Y." a.k.a. "R U Dead Yet?" [63].

3.2. Generic DoS Defense and Mitigation. In operational settings, several traditional DoS defense strategies have been adopted for generic, non-cloud networks in order to minimize impacts of volumetric and protocol DoS attacks. For example, usage of firewalls and filtering can help mitigate DoS attacks by dropping malicious traffic and control what traffic can reach the infrastructure. However, firewalls can also lead to false positives, i.e., filtering out legitimate packets. Moreover, firewalls can be susceptible to high-volume flood attacks since firewalls' state tables can only hold a certain number of sessions. For some particular attacks, disabling or limiting some functionalities can help prevent DoS. For example, disabling UDP support by default is a good method to cope with Memcached amplification attack or reducing the number of open DNS resolvers can help limit DNS amplification attack. Intelligent routing and diversion techniques such as using content distribution networks (CDNs) [64] or load balancers [65] can help break the massive traffic into manageable chunks as well as prevent direct traffic to important parts of your system. Although this approach is very effective, it requires a lot of resources and therefore may not suitable for resource-constrained environments. When it comes to application attacks, due to their complexity, defenders usually have to combine multiple defense methods such as firewalls, pattern adaptation, and even incoming requests rate limiting in order to be effective [66, 67]. Nevertheless, in many cases, these traditional operational methods fall short; thus, new intelligent approaches have been proposed.

One such approach is packet payload intervention at servers or routers [68–71]. Server side intervention of SYN cookies [68] is a popular method to fight SYN flood attack. Here, upon sending SYN-ACK packet back to the client, the server drops the original SYN request from the queue. If the ACK message eventually arrives, the server rebuilds the SYN packet using a cryptographic technique. Consequently, there always remain available ports for new handshake establishments and thus new connections are not denied. For

router-side intervention, works such as [69, 70] try to manipulate the essential information inside the packet payload to come up with defense strategies. In [69], the authors proposed router stamping that helps identify the source the DoS attacks hidden with IP spoofing [54]. If a packet travels via three routers, each router will record the IP address of its predecessor before it forwards the packet. By counting the number of stamped packets at each router during an attack, the routers can anticipate the source of the attack. In [70], the authors proposed DoS defense, namely, NetFence at bottleneck routers, i.e., the routers at service provider side that are responsible for inbound traffic. Bottleneck routers stamp the packets that carry congestion monitoring feedback to signal congestion to access routers, while other access routers use it to monitor senders' traffic. These congestion monitoring feedbacks are encrypted so that they cannot be faked.

Another way of payload modification for DoS defense is implementation of pushback that was first presented in [71]. The pushback method considers a DDoS attack as a congestion problem by dropping the traffic at the congested points and propagating the information back to upstream routers in order to force them to rate-limit the traffic, i.e., pushback. The authors also proposed a heuristic algorithm to filter bad traffic that further improves pushback mechanism. Nevertheless, the payload modification approach has some limitations. Firstly, it requires cooperation between router manufacturing companies such as Cisco [72] and Juniper [73] or software platforms such as Linux Foundation [74] or FreeBSD Project [75] in order to make the modified packets compatible with router hardware and drivers. Secondly, this approach has to monitor and/or modify the packet payload which could add significant overhead on packet encapsulation and decapsulation process and certain inaccuracies (e.g., SYN cookies).

Another intelligent approach, namely, honeypot [76–78], has been very popular in mitigating DoS. Honeypot is a decoy system designed to act like a real system with data and resources having no legitimate use [76, 77]. It is typically set outside the internal network in order to lure attackers to perceive it as the real system. Most often, honeypot is configured as part of the most external layer of the network or the science DMZ [14]. That way, if the network is under attack, honeypot will be hit first. For sophisticated attacks, it is possible that both the honeypot and the network/server are attacked simultaneously [76]. However, the honeypot can help in quick analytics on the attack traffic and use that information for recovery and/or quarantine [76, 78]. In works such as [78], honeypots are even used to identify the infrastructures behind DNS amplification attacks. Although effective, honeypots are not designed to mitigate DoS attacks, rather to act as decoys for analytics and information collection. Furthermore, if not properly deployed, honeypot could attract unwanted attack traffic which could lead attackers to penetrate the internal networks [76]. Although these methods can be applied to cloud environments, due their bigger scale and existence of new threats, cloud environment inspired new approaches towards DoS defense.

3.3. Traditional DoS Defense in Cloud Environments. DoS attacks targeting cloud services and infrastructure also fall under the aforementioned three categories, i.e., volumetric attacks, protocol attacks, and application attacks. For DoS defense strategies in cloud, based on the focus of this survey we categorize them into two categories: traditional or non-MTD-based and MTD-based. The MTD-based defense strategies will be discussed in Section 4. Here we introduce the traditional or non-MTD approaches for DoS defense in cloud infrastructure. Broadly many DoS defense strategies applied to non-cloud infrastructures can be borrowed for cloud environments. However, many authors have proposed new methods that leverage the uniqueness of cloud infrastructure such as softwarization, virtualization, and elasticity (e.g., on-demand). For DoS/DDoS defense and mitigation designed for cloud infrastructures, works such as [79–92] are notable that can be broadly categorized into groups shown in Figure 6.

3.3.1. Leveraging SDN and Virtualization. These groups of works [79–82] use the programmability and virtualization of SDN-enabled cloud infrastructure to defend against DoS/DDoS attacks. The authors in [79] implemented a cloud-based overlay network (i.e., a virtual network built on top of physical networks) that provides an integrated set of on-demand security services such as intrusion detection systems (IDSs), DDoS prevention, and firewalls. In [80], Fayaz et al. presented Bohatei, a flexible and elastic system that leverages SDN and virtualization with a resource management algorithm to drive malicious traffic through the defense system while minimizing latency and network congestion. In [81], Zhang et al. proposed Poseidon, a volumetric DDoS defense strategy that is adaptable to attack patterns and leverages SDN's programmable switches combining the advantages of hardware-based and software-based defenses. Similarly, Liu et al. in [82] took advantage of SDN's programmable switches to introduce Jaqen, a programmable switch-native tool that can run detection and mitigation functions without relying on additional hardware.

3.3.2. Anomaly Monitoring and Detection. This group of works [83–85] proposed anomaly monitoring and detection strategies to segregate anomalous traffic. Narayana et al. in [83] proposed a SDN-based path query language for efficient path-based traffic monitoring that can help measuring the flow of traffic, which is crucial for many tasks, including DoS mitigation. Elsabagh et al. in [84] proposed Cogo, a proactive probabilistic system for early detection and mitigation of application DoS attacks such as low-and-slow attacks. In [85], Demoulin et al. introduced FineLame, a framework for detecting asymmetric DoS attacks (attacks that target applications' internal algorithms or semantics) via resource monitoring.

3.3.3. Intelligent Routing and Diversion. These works [86–88] propose intelligent routing and subsequent diversion techniques to isolate attack traffic. Works such as [86]

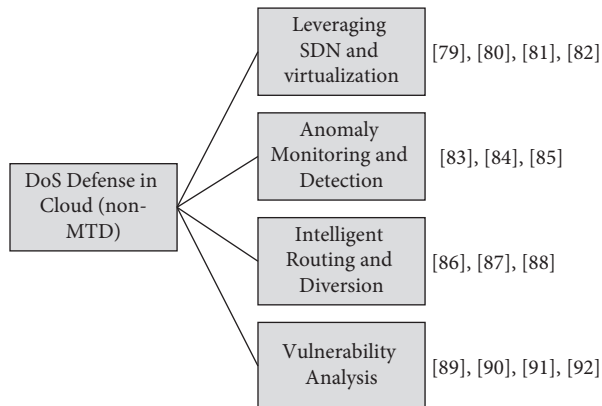


FIGURE 6: Categories of traditional DoS defense strategies in cloud infrastructures.

leverage the flexibility of cloud resources to deploy an affordable CDN-based solution, namely, CDN-on-demand to mitigate volumetric DoS attack and flash crowds. In [87], Ramanathan et al. proposed SENSS, a security service that can help the victims to ask the upstream Internet service providers (ISPs) for help by requesting on-demand attack monitoring and filtering. The authors in [88] proposed DynaShield, an on-demand low-cost crypto-based solution that can auto-scale to large attacks to cope with volumetric DDoS attacks.

3.3.4. Vulnerability Analysis. Unlike other works that propose solutions based on existing DoS vulnerabilities, this final group of works [89–92] investigates new threats and vulnerabilities and measures weaknesses and strengths of cloud-based solutions using case studies. In [89], Vissers et al. investigated attack vectors in which attackers can exploit to discover the IP addresses of important parts inside the infrastructure of cloud-based security providers and also evaluate their impacts. In [90], Bushart and Rossow presented DNS unchained, a new amplified application-based DoS attack against DNS authoritative servers and its impacts. In [91], Jansen et al. investigated a volumetric DoS case study against Tor anonymity network [93] via some default Tor bridges that reside on popular CSPs. Kopp et al. in [92] investigated the impact and anatomy of booter-based DDoS. Booters are DDoS-as-a-service providers that offer their customers DDoS services for an affordable price.

It is important to notice that most of these works focus on volumetric and application attacks, while there exists very few novel work on protocol attacks. This is quite understandable because to perform a successful protocol attack, the attackers first have to discover a network or transport layer protocol vulnerability and then exploit that. Since there are limited number of de facto and standardized network and transport protocols on the Internet and most of their vulnerabilities have been discovered and patched, there is not much left to exploit. Further proof for this is that most protocol attacks such as SYN flood, ping of death, Smurf, and part of IP fragmented attacks have been considered largely solved with newer software updates. However, the

advent of SDN and OpenFlow has significantly changed the landscape. Although it opens a broader scope of DoS defense in cloud, SDN and OpenFlow also come with their own vulnerabilities [94–96]. One of these new vulnerabilities in OpenFlow protocol have been exploited to launch a reflection-based attack, viz., table-miss [97–101] that can completely cripple both the switches and the controller.

4. MTD for DoS Mitigation in Cloud

In order to address the rapid growth of DDoS attacks, the cloud security community and federal organizations are exploring “Cyber Agility and Defensive Maneuver (CAADM)” mechanisms for cloud that can allow for real-time service restoration through agile cloud resource adaptation once an attack is detected and also limit proliferation of detected attacks within the cloud environment through preventive maneuvers [30]. In order to realize such CAADM mechanism in cloud, MTD-based techniques are the need of the hour [31], as (1) intelligent but fast converging algorithms can be developed for both proactive and reactive maneuvers based on triggers for a wide range of global and local greedy optimization criteria; (2) emerging network management technologies such as SDN can help implement and operationalize such dynamic and agile maneuvers in order to evade impending attacks; and (3) sophisticated dynamic maneuvers can be designed to create system obfuscation helping to deceive/illude the adversary in a false sense of success and thus stopping the proliferation. In this survey, we broadly categorize the current research landscape in MTD-based DoS defense for cloud environments into the following three categories based on the adopted MTD-based maneuvering mechanism, viz., network address shuffling-based, proxy-based, and live migration-based. Then for each category, we further subcategorize the works into the following two groups based on the adoption of SDN or other programmable technologies for MTD implementation, viz., non-SDN MTD and SDN-enabled MTD. The overall classification is illustrated in Figure 7. Below we discuss the theoretical and system design details for each such category. The evaluation techniques adopted for each such work and corresponding results are later discussed in Section 5.

4.1. Network Address Shuffling. Network address shuffling and randomization is the classic approach and one of the most popular implementations of MTD in cloud. In this technique, network addresses (e.g., IP addresses) associated with the application servers or virtual machines (VMs) are reassigned or randomized around an available pool of addresses (e.g., from DNS servers) periodically (can be fixed or adaptive). The cloud service users who are oblivious to such randomization are then redirected to new IP addresses without significant quality of service (QoS) drop (Figure 8). Such randomization considerably increases attacker cost, and it has to continuously guess the network addresses or address space associated with the target server or VM. In recent times, SDN-enabled shuffling and randomization

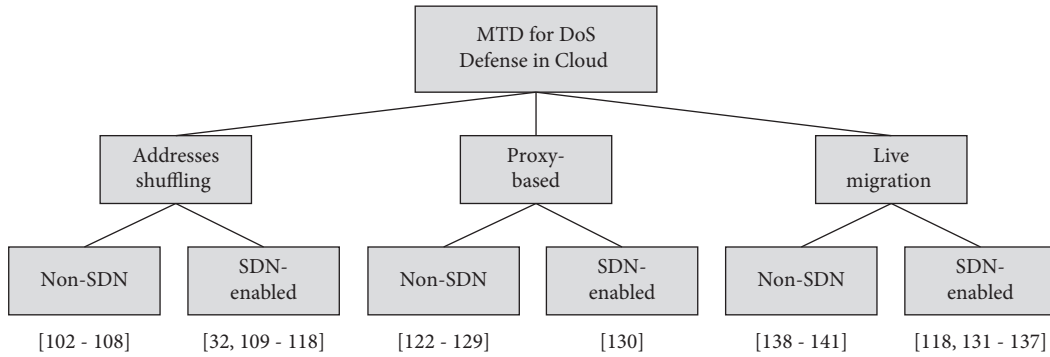


FIGURE 7: MTD for DoS defense in cloud environments.

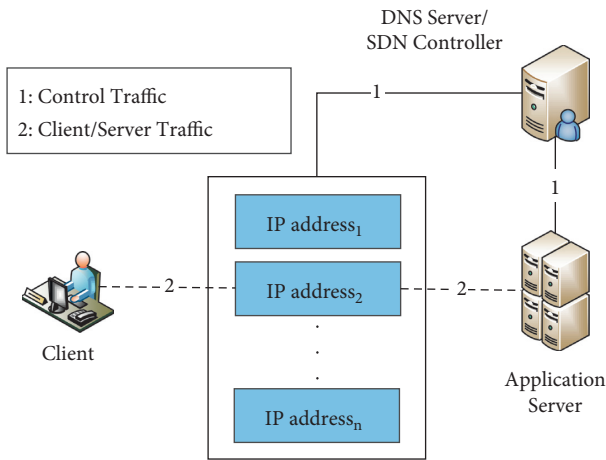


FIGURE 8: Logical diagram of MTD implementation through IP shuffling.

work are gaining momentum. In most cases, such implementations are reactive in nature, i.e., the defense scheme kicks in once an attack is detected. However, SDN can also be useful in implementing such maneuvering proactively thanks to the complete centralization of the network control plane and can help prevent impending attacks. The usage of the decoupled SDN controller allows easy deployment of monitoring, predictive, and defensive algorithms that work in complete harmony. Among the works that employ network address shuffling, [102–109] are notable for non-SDN-based methods, while works such as [32, 110–117] propose SDN-enabled shuffling and randomization techniques.

4.1.1. Non-SDN Implementations. Among these works, Carroll et al. in [102] presented probabilistic models for IP address shuffling-based MTD. These models quantify the attacker success under different conditions such as network size, number of addresses, and number of vulnerable systems. The authors investigated the relationship between shuffling frequency and connection loss and found that shuffling provides limited protection against attackers focusing on one high-value system. Their results also indicate that shuffling is acceptable if there is a small pool of vulnerable systems within a large network address space but may cost connection losses of legitimate users. In [103, 104],

Alavizadeh et al. investigated the effectiveness of individual shuffle, diversity, and redundancy-based MTD techniques and proposed a method that combines all three. They use a graphical security model, viz., hierarchical attack representation model (HARM) [118], to model and analyze the MTD techniques. Wang et al. in [105] studied the MTD timing problem, i.e., the optimal time to conduct the adaptations and to balance the cost-effectiveness. The authors devised a multimodule framework along with a cost-effective adaptation algorithm called renewal reward theory-based solution (RRT) to cope with this issue.

In [106, 107], Clark et al. presented a game-theoretic framework that combines decoy network and address space randomization to distract and mislead adversaries. The proposed framework consists of two components: (i) one that differentiates between the decoy nodes interactions with the adversary and a real node and (ii) another for adversarial game formulation in order to find the attack target in a network consisting of real and decoy nodes. Moreover, the authors argued that the designed framework only needs to randomize IP addresses if the adversarial scanning rate exceeds a certain threshold. Nizzi et al. in [108] presented a cryptography-based address shuffling algorithm, namely, ASHA, for IoT devices in wireless sensor networks (WSNs) to deal with security threats including DoS attacks. The proposed algorithm uses address renewal methods by leveraging cryptographic hash functions that aims to be simple and collision-free and have low overhead. Likewise, in [109], Yao et al. also proposed a network address shuffling approach for IoT devices to eliminate security threats in WSN. The authors formulated the problem as a stochastic cost optimization problem and proposed a novel stochastic cost minimization mechanism (SCMM) to solve it.

4.1.2. SDN-Enabled Implementation. Among these works, Kampanakis et al. in [110] analyzed how SDN can be used for MTD by investigating the advantages and disadvantages of network-based MTD techniques. The authors argued that programmability with SDN controller can help with system/resource adaptations which is an important factor to MTD maneuverability. Also, a highly programmable SDN-based system can provide obfuscations that will increase the cost of an attack by making the attacker spend more resources in order to study the attack surface(s). Steinberger et al. in [111]

investigated how MTD can leverage SDN to the fullest extent. The authors argued that if MTD strategies are implemented using SDN in a collaborative environment, the impact of large-scale DDoS attack can be significantly reduced. They argued that MTD can limit the attacker's knowledge of the target due to the ever-changing attack surface (because of MTD) and thus can increase attacker cost. The advantage of their collaborative DDoS defense solution is that using their system, each participating partner achieves insights into the current threat landscape. Further, collaborative DDoS defense pools expertise and resources from all collaborating partners, thus achieving greater success against attacks. This work also indicates that ONOS [119] is an appropriate SDN OS to enforce implementation of MTD due to its guaranteed scalability as ONOS has been used and tested in several high-speed networks. In [112], Zhou et al. proposed a new cost-effective shuffling (CES) method against DDoS attacks using MTD based on game theory. CES takes shuffling frequency into account and models the interaction between the attacker and defender using multiobjective Markov decision processes. Based on this model, the authors studied the best trade-off between the effectiveness and cost of shuffling in each particular scenario.

Jafarian et al. in [113, 114] proposed an address randomization technique called random host-address mutation (RHM) to mitigate reconnaissance attacks. This technique can turn the servers into untraceable moving targets by leveraging SDN to mutate their original network addresses. The actual IP addresses (rIP) are kept unchanged, but it can create routable short-lived ephemeral IP addresses (eIP) from the unused ranges of the network address. The eIP addresses are provided via DNS and are used for routing. They are automatically translated back into the rIPs and vice versa at the network edges close to the destination. RHM utilizes a two-level mutation scheme to maximize the unpredictability: (i) low-frequency mutation (LFM) that changes the set of unused ranges assigned to each host and (ii) high-frequency mutation (HFM) that assigns the new eIP address associated with each host.

In [32, 115], Chowdhary et al. sought to tackle DDoS attacks by selecting suitable countermeasure based on obtained information about the adversaries. However, the authors chose two different paths to obtain the needed information. Work in [32] presents an automated dynamic system reconfiguration by leveraging scalable attack graphs (AGs) to assess the attacks and select necessary countermeasures to perform real-time network reconfiguration, both proactively and reactively. A node in an AG is a combination of hosts and the possible vulnerabilities that exist on that particular host. Each host may have intra-connections or interconnections with other hosts. Hence, if a botnet communicates with clients to target a system resource, this information can be modeled and tracked. This scheme also ensures that there is no security policy violation or conflict after the adjustments are done, whereas in [115], the authors combined SDN-enabled MTD with the intrusion detection system (IDS) to formulate a threat scoring system based on vulnerabilities and IDS alerts and selected MTD

countermeasure. This defense mechanism is called MASON, and instead of IP addresses, it uses the port hopping technique. Based on threat scores, MASON can identify network services with high-security risk and take corresponding actions.

Aydeger et al. in [116] presented a signaling game to thwart the emerging crossfire attack, a type of stealthy link flooding attack (SLFA). It is a variant of DDoS attacks that congests the connections surrounding the network of the target servers by sending low-volume traffic from many bots. The proposed signaling game considers the defender and the attacker as two players, and the equilibria represent the best strategies for each player. Based on the game results, the authors proposed an improvement upon random route mutation (RRM) [120], viz., strategic RRM. It is a multipath routing algorithm that periodically changes routing to avoid passing through some compromised links or nodes [121]. Similarly, Xu et al. in [121] also proposed an improvement over route mutation algorithm for MTD. They modeled route mutation process as a Markov decision process and introduced a context-aware Q-learning RM algorithm (CQ-RM) that can learn attack strategies to optimize the selection of mutated routes adaptively.

In [117], Nguyen et al. proposed Whack-a-Mole, a SDN-driven MTD mechanism for DDoS defense in cloud environments. Whack-a-Mole resource maneuvering works at two levels: (i) it proactively spawns replicas of VMs hosting critical applications where the applications are seamlessly migrated and (ii) it mutates the IP addresses associated with the services by assigning the VM replicas with IP addresses belonging to different address spaces (assuming that the entire cloud network is divided into different address spaces). Upon resource maneuver, the OpenFlow switches with the help of SDN controller direct all new incoming user requests to the spawned VMs, whereas the existing users are allowed to finish their sessions with the old VMs. Upon completion of the existing users' sessions, the VMs are terminated and IP addresses are recycled for newly spawned VMs. In their work, the address mutation is optimized to keep the new IP address selection as unpredictable as possible to increase attacker cost.

4.2. Proxy-Based. In this method, the IP address of the real server or VM (which in most cases is the target) is concealed from all clients and the real servers hide behind a group of intermediate proxy machines or VMs. Clients first communicate with the control unit (e.g., authentication server) that directs them to the correct proxy. The MTD-based maneuvering is initiated periodically (fixed or dynamic) when the physical and/or logical identity of the proxy that is being connected to the real server is changed to another (as shown in Figure 9). The identity of the new proxy can be random or based on some intelligent mechanism. Thus, for the attacker trying to target a server or VM, figuring out the identity of the proxy is essential and for obvious reasons non-trivial. Unlike other categories of MTD works, most of the current state-of-the-art proxy-based MTD techniques such as [122–129], except [130], can be implemented

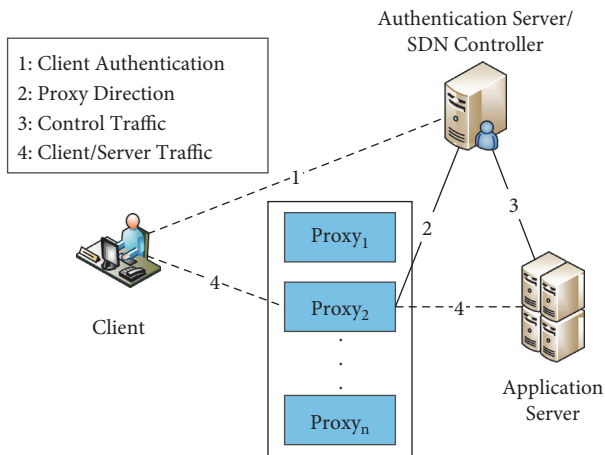


FIGURE 9: Logical diagram of MTD that implements proxy-based mechanism.

successfully in spite of not having SDN-like programmability in the system.

Jia et al. in [122] and Wang et al. in [123] proposed MOTAG, a proxy-based MTD mechanism that utilizes a layer of secret moving proxies to mediate all communications between the clients and the protected VMs. The filters deployed surrounding the VMs only allow traffic from the valid proxy nodes. The proxy system acts as a shield between the VMs and the rest of the Internet. When one proxy node is under attack, it is replaced by another node at a different network location and the associated clients are redirected through the new proxy. With the proposed algorithms, the proxy nodes can also be used as an isolated environment for the potentially malicious users working as insiders. Similarly, Wood et al. in [124] devised a relay network called DoSE that acts as a proxy between clients and servers. The relay node is located in the public cloud infrastructure, and content delivery networks (CDNs) help to disseminate the relay information to the corresponding clients. DoSE aims to achieve low-cost DDoS attack mitigation for small to medium-sized organizations that typically have limited budgets. DoSE connects clients to relay proxies and proposes new methods for assigning clients to relays in order to mitigate network layer attacks while minimizing costs.

Fleck et al. in [125] and Kesidis et al. in [126] extended the work on MOTAG and utilized it to proactively minimize DDoS attack's impact by attempting to thwart potential attacks during the reconnaissance phase. The authors studied a proactive and cloud-side MOTAG defense in which proxies dynamically change to thwart DDoS attack's reconnaissance phase and consequently reduce the attack's impact. In these works, the authors used a load balancer to direct clients to the proxies. They used an adversarial coupon collection-based mathematical model to formulate the problem. In [127], Bandi et al. also presented a MOTAG-like strategy combined with Fast-Flux, a technique used to hide the servers behind an ever-changing system of proxy. The authors proposed FastMove, a shuffling algorithm to determine the number of legitimate clients on each proxy server in order to save the largest possible number of clients.

In [128], Venkatesan et al. argued that proxy-based defense mechanisms such as MOTAG and DoSE can be vulnerable to a new type of attack, namely, the proxy harvesting attack. The proxy harvesting attack exploits a weakness in the authentication process of these proxy-based architectures to collect information about a possibly large number of proxy nodes with the help of insiders. To overcome the proxy harvesting attack, the authors proposed BIND-SPLIT strategy that limits the number of IP addresses that can be harvested, combined with the proactive defense mechanism called PROTAG. The proposed PROTAG mechanism helps with two primary factors: (i) proxy selection to determine the optimal proxies to be replaced and (ii) movement frequency to determine the optimal time to replace the proxies.

In [129], Wright et al. introduced a novel game-theoretic model that formulates a DDoS attack as a two-player normal-form game between the attacker and defender. In this game, both sides want to affect the quality of experience (QoE) of the legitimate clients, while keeping their own costs low. This work is called MOTAG game as it is built upon the MOTAG model with an objective to evaluate the effectiveness of proxy-based MTD strategies. To achieve that, the authors used the simulation-based empirical game-theoretic analysis (EGTA) to find game-theoretic equilibria in complicated games over restricted strategy spaces.

As mentioned earlier, most of the proxy-based MTD techniques are implemented without having SDN-like programmability in the system, except [130]. Here Aydeger et al. proposed a shadow network (SN) framework to deal with crossfire attack (this work shares the same goals as in [116]). SNs are tiny low-cost networks (can be virtual or physical) attached to actual ISP network which are used to deceive attackers with the fake topology information. In this proposed framework, the ISP network is assumed to be SDN-based through which the ISP can perform traffic engineering to any traffic flow using the SDN controller. The location of SNs is similar to proxies as they reside between the protected servers/VMs and the clients. Nevertheless, the SNs do not completely shield the entire internal network like the proxies but are rather a part of it.

4.3. Live Migration. In this method, a pool of VMs working as application servers (these VMs can be created beforehand or on the go) is responsible for hosting the target services. In order to create system obfuscation, periodically (can be fixed or dynamic) the services are migrated from one set of VMs to another by a control server as shown in Figure 10. The VMs currently not hosting any services are kept on standby and ready to host at a moment's notice. Such migrations are often called live migrations as all the migration-related actions such as taking a VM snapshot and transferring the files from one VM to another happen "online" while the users are still using the services. The optimal VM or set of VMs selection for migration can be based on complex algorithms that consider factors such as VM's capacity, current specifications, and network address space where it resides among many others. After the migration is complete,

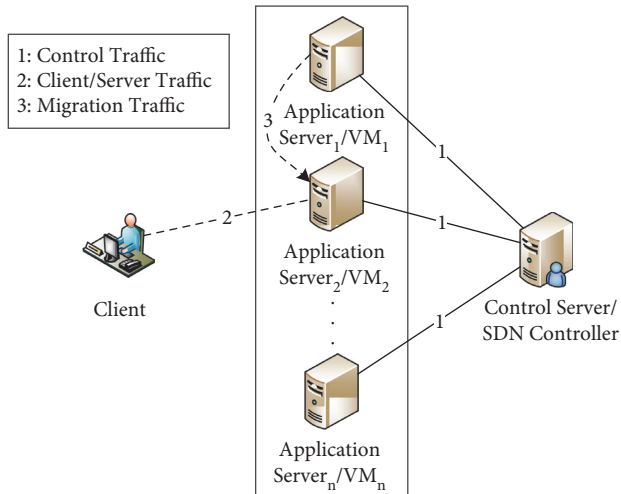


FIGURE 10: Logical diagram of MTD implementing live VM migration and user redirection.

the users of the service are redirected to the new VM(s). This considerably increases the attacker cost as the attacker has to first identify the VMs currently hosting the target services before launching an attack. In many of the current research studies [117, 131–133], live migration and user redirection are often carried out by the centralized SDN controller. The decoupled and centralized SDN controller provides added flexibility and dynamicity to apply a host of intelligent algorithms that are effective against impending attacks. However, there also exist a number of research works [134–136] that achieve VM migration using traditional non-SDN-based methods.

4.3.1. Non-SDN Implementation. Among the works that propose traditional non-SDN-based migration, Jia et al. in [134] proposed a cloud-enabled, shuffling-based MTD strategy to marginalize the attackers within a space of VMs. When under attack, it replicates the attacked VM instances, migrates it to a newly instantiated replica VM at a different network locations, and assigns legitimate clients to the newborn VM. In order to prevent moving sophisticated attackers to the new replica VMs, the authors keep track of the legitimate client assignments and if the new replica is attacked, they separate benign client sessions from potentially malicious ones. Through multiple rounds of shuffling, they can filter out the attackers and enclose them. The authors also introduced a novel family of algorithms to optimize the mitigation runtime and minimize the number of shuffles.

Peng et al. in [135] modeled a cloud-based system with heterogeneous resources and dynamic attack surfaces to ascertain whether and to what extent MTD is effective. The authors used a VM migration technique that moves the snapshots of servers within the pool of VMs. The novelty of this work is its consideration of the attacker’s accumulated knowledge about the attack surface and formulation of a stochastic problem that wants to minimize the probability of the service being compromised.

In [136], Venkatesan et al. proposed a MTD approach to defend against stealthy botnets for resource-constrained environments. The authors deployed detectors across the network and applied a series of defense strategies to periodically change the placement of those detectors. The objective is to make attackers uncertain about the location of detectors so that they have to perform additional actions in an attempt to create detector-free paths through the network, hence increasing the attackers’ likelihood of detection.

4.3.2. SDN-Enabled Implementation. Among the works that implement VM migration using SDN, Debroy et al. in [131, 132] proposed a DDoS defense mechanism that allows for proactive migration of target application for impending attacks and triggers reactive migration when under attack. This work’s novelty is in the moving frequency optimization and the ideal location selection for migration across heterogeneous pool of VMs based on attack probability. The objective for such optimization is to make the migration frequent enough to evade impending attacks at the same time not too frequent that it causes unnecessary resource wastage. In order to find the ideal VM, the authors proposed an optimal market-driven approach that is based upon distributed optimization principles. This approach uses virtual market economics in order to optimize resource allocation during migration. Besides, as part of the reactive defense, the authors also presented false reality scheme that reuses the attacked VM as a trap to deceive the attacker and gather adversarial information.

In [133], Nguyen et al. proposed a common vulnerability scoring system (CVSS) [137] driven Bayesian attack graph (BAG) model designed for low-budget and small-scale private cloud infrastructures such as campus private clouds (CPCs). This model is used to perform a dynamic threat/risk assessment for integrity, confidentiality, and availability attacks on data residing in the VMs. BAGs are used to model cyberattack causal relationship and used for assessing attack success likelihoods. The likelihood of an attack success is calculated using CVSS. As for the vulnerabilities, the authors used relevant cyberattack statistical data from Common Vulnerabilities and Exposures (CVE) [138]. Using the proposed model, they performed a case study on the campus network to evaluate the likelihood of success of confidentiality, integrity, and availability attacks with and without MTD-based maneuvering. Finally, in Whack-a-Mole [117], Nguyen et al. proposed a live VM spawning and migration scheme where replicas of VMs hosting critical applications are proactively spawned and where the cloud-hosted applications are migrated to. Once a VM is spawned, all the new service requests are routed to the newly spawned VM, whereas the old VM only services the existing users. Each VM has a lifetime after which its resources are reclaimed. This lifetime is optimized to serve the average span of a service request session in order to minimize user QoE degradation. At the same time, the authors optimized the spawning frequency that is large enough to thwart an impending attack, yet not too large that it causes unnecessary resource wastage from too frequent spawning.

A comparative summary of the aforementioned different MTD strategies and their relative pros and cons against common types of DoS attacks (discussed in Section 3) is described in Table 3.

5. Evaluation Methodologies

In this section, we discuss and compare the evaluation methodologies used in MTD-based DoS defense for cloud systems discussed in Section 4. In order to broadly capture the different types of evaluation strategies used in such works, we categorize them into three groups: simulation-based, hardware testbed-based, and cloud testbed-based, as shown in Table 4. For each of these groups, we discuss how the related works evaluate the most important metrics to gauge the proposed strategies' success, viz., security and usability. As shown in Figure 11, security metrics are typically evaluated using attack probability, attack graphs, and risk computation, whereas most important and relevant usability metrics to be evaluated are cost and performance. Figure 11 also illustrates the list of works corresponding to each such metric.

5.1. Simulation-Based. In the absence of hardware testbeds and cloud testbeds, simulation-based (sometimes numerical) approach is a good first step towards evaluating the success of the proposed strategies. In this method, the authors typically use software platforms such as MATLAB [140] and available datasets to simulate their methods and models. Due to the lack of real experiment setups, the authors mostly use this method in analysis work and rarely use it alone to evaluate system performance. Simulation results are typically used along with the other two evaluation methods. Works such as [102–104, 106–109, 117, 121–124, 126, 128, 129, 131–136, 139] use extensive simulation results to demonstrate system performance.

5.1.1. Security Metrics. For attack probability approaches, the authors in [102] constructed a probability model to measure the mean time to security failure where the security failure is defined by the system state being compromised by an attacker and where the system is defended by MTD-based maneuvering. In works such as [117, 131, 132], the authors simulated their Poisson point process-based model to measure the optimal moving interval to minimize the probability of getting hit by the attacks. They simulated the model for different attack budgets in terms of the ratio of attack time and idle time. The authors in [113, 114] developed probabilistic models to measure attack probability when a set of reconnaissance defenses that includes deception technique and network address shuffling as MTD is deployed in a given system, while varying the network size, the size of VM deployment, and the number of vulnerable nodes. The authors set up a virtual network using Mininet [141, 142] and a SDN controller. The evaluation shows that RHM provides a robust performance in countering sophisticated threat models for both proactive and adaptive schemes with low overhead. In [108], the authors provided

probabilistic models to measure the effectiveness of an address shuffling-based MTD technique with respect to the network size, the address space scanned, the degree of system vulnerability, and the frequency of shuffling operations. The results indicate that for a typical personal area network (PAN), ASHA can effectively mitigate DoS attacks with tunable overhead.

Among works employing attack graph approaches, Nguyen et al. [133] simulated all combinations of an attack graph based on the BAG model. They compared the attack success rate for confidentiality, integrity, and availability attacks for both MTD-based and non-MTD-based approaches. In [136], the authors proposed a technique to capture the dynamic changes in the network resulting from deploying MTD during the entire simulation runtime. Using two metrics, viz., minimum detection probability and attacker's uncertainty, the simulation results show that the proposed approach can effectively reduce the likelihood of successful attacks. Work such as [103, 104] use the hierarchical attack representation to model a system's security features with two layers, an upper layer and a lower layer. The upper layer represents a network's reachability information (i.e., network topological information) while the lower layer represents a node's vulnerability information using attack graphs. The results show that the proposed combined techniques can satisfy the evaluation criteria while individual techniques do not. The authors in [32] also analyzed the main advantages of using attack graphs, viz., ease of evaluation and representation. Furthermore, attack graphs can be adopted to compute various security metrics based on the MTD application.

Among works employing risk assessment approaches, the authors in [106, 107] provided new metrics for MTD evaluation and risk analysis. They proposed statistical metrics to study the effect of how the attacker can quickly conduct and succeed in adversarial attacks. The authors assumed that the system will always have a running task that can be measured. The results show that networks should consist of a mixture of high-interaction (that implements the full protocol) and low-interaction (that implements a subset of protocol states) VMs. In [110], the authors considered game theoretical formulation of MTD systems; specifically, they modeled it as a Markov game. The authors provided a theorem, subject to probabilistic constraints, to calculate the revenue for the defensive and offensive approaches in MTD systems. Their work depends on testing different defensive and offensive strategies and is tested using a networking setup that includes vulnerable services and a firewall component.

5.1.2. Usability Metrics. Among these, works [115, 121, 139] used some form of cost function for the evaluation. The authors in [121] aimed to identify an optimal interval of VM migration in order to maximize security with minimum cost based on a game-theoretic formulation called Vickrey–Clarke–Groves (VCG) mechanism. The simulation results show that proposed mechanism provides significant improvements in multiple aspects including defense, mutation

TABLE 3: Summary of different MTD strategies and their relative utility against common DoS attacks.

Approaches	Features	Performance against volumetric attacks	Performance against protocol attacks	Performance against application attacks
Network address shuffling [32, 102–118]	<ul style="list-style-type: none"> (i) A pool of network addresses (e.g., IP addresses) is managed dynamically. (ii) Network addresses of the target server(s) are reassigned or randomized periodically. (iii) Requires less physical resources. 	<ul style="list-style-type: none"> (i) Works for most volumetric attacks because the targets are masqueraded under different network addresses. (ii) Not effective against DNS amplification attacks as IP addresses are resolved during attacks. 	<ul style="list-style-type: none"> (i) Works for most protocol attacks because the targets are masqueraded under different network addresses. 	<ul style="list-style-type: none"> (i) Does not work for application attacks due to the fact that such attacks target domain names rather than network addresses.
Proxy-based [122–130]	<ul style="list-style-type: none"> (i) The target server(s)' identities are concealed from all clients behind a group of intermediate proxies. (ii) The identities of proxies can be static or dynamic. (iii) Requires more compute and network resources for the pool of proxy servers. (iv) For effective implementation, it may need support from other tools such as firewalls. (v) Proxies can also help in early detection of attacks. (vi) Using proxies can open up other vulnerabilities [128]. 	<ul style="list-style-type: none"> (i) Works for volumetric attacks as the targets are protected behind proxies. (ii) The proxies act as the first line of defense and face the brunt of the attack. 	<ul style="list-style-type: none"> (i) Works for flooding-based protocol attacks (e.g., SYN flood) as the targets are protected behind proxies. (ii) Does not work for stealthy protocol attacks (e.g., IP fragmentation) that can percolate through the proxies, even when they are SDN enabled. 	<ul style="list-style-type: none"> (i) Works for flooding-based application attacks (e.g., HTTP flood) as the targets are protected behind proxies. (ii) Does not work for low-and-slow protocol attacks that can percolate through the proxies, even when they are SDN enabled.
Live migration [118, 131–141]	<ul style="list-style-type: none"> (i) A pool of physical/virtual resources is kept on standby to host target services. (ii) The services are migrated to and from these resources. (iii) Such migrations can be proactive (i.e., periodic) or reactive (i.e., when under attack). (iii) Such redundancy typically requires resource abundance. (iv) Often requires SDN-based implementation with other strategies (e.g., proxy-based) to be effective. 	<ul style="list-style-type: none"> (i) Works for volumetric attacks as the targets are moved around proactively using a SDN controller. (ii) Works even when there is only a reactive scheme (with or without SDN) where the target can be quickly migrated to safety. 	<ul style="list-style-type: none"> (i) Works for flooding-based protocol attacks (e.g., SYN flood) as the targets are moved around with or without SDN. (ii) Works for stealthy protocol attacks (e.g., IP fragmentation) as long as the attacks are detected early and SDN migrates the target(s) rapidly. 	<ul style="list-style-type: none"> (i) Works for flooding-based application attacks (e.g., HTTP flood) as the targets are moved around with or without SDN. (ii) Works for low-and-slow protocol attacks as long as the attacks are detected early and SDN migrates the target(s) rapidly.

TABLE 4: Categories of evaluation methodologies used in MTD-based DoS defense works.

Groups	Simulation-based	Hardware-based	Cloud-based
Works	[102–104, 108]	[105, 110–112]	[125, 128, 134]
	[106, 107, 109]	[113–116]	[131, 132]
	[121–123, 126]	[127, 130]	[117, 133]
	[124, 128, 129]		
	[117, 131–136, 139]		

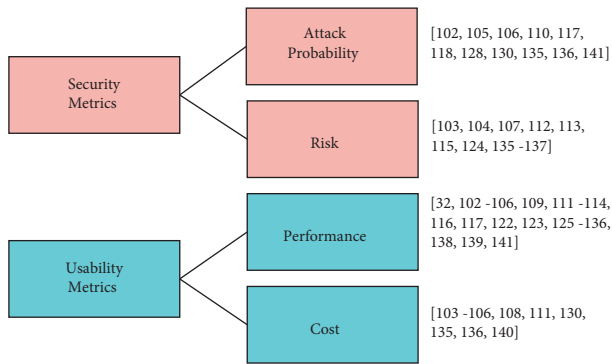


FIGURE 11: Security and usability metrics used in related works.

overhead, network, and convergence performance compared to state-of-the-art methods. In [139], the authors evaluated the system cost caused by an attacker at different stages of the network. The attacker-defender game is modeled as a finite zero-sum matrix game with a bounded cost function with a mixed-strategy saddle point equilibrium (SPE). Players utilize cost function learned online to update MTD strategies. The numerical results show that feedback mechanism allows network defense to respond to unexpected events.

Among works that evaluate performance as a usability metric, the authors in [113, 114] identified the virtual IP (vIP) mutation, range allocation, and range distribution constraints in order to minimize the quality of service (QoS) impact induced by vIP collisions as well as to maintain optimal level of unpredictability. Probabilistic performance analysis of MTD reconnaissance defense was conducted in [139]. This work analyzes quantifiable MTD metrics such as reconnaissance, deception performance, attack success probability vs. connection drop probability, and attacker's success probability under different conditions such as network size and number of vulnerable computers. The authors in [115] used mission and attack metrics for analyzing the effectiveness of network defense. They analyzed dynamic defenses such as "Active Re-Positioning in Cyberspace for Synchronized Evasion and Self-Shielding Dynamic Network Architecture" using mission and adversary activity set. Here mission success, i.e., the rate at which mission tasks are completed, and mission productivity, i.e., how often are mission tasks successful, are used as QoS measurement metrics for evaluations. In [135], via simulation, the authors identified the conditions and extent of the proposed strategy's effectiveness. From the results, they concluded that (i) VM migration is more effective when the pool of VMs is dense and/or when the attack is large scale and (ii) the heterogeneity and dynamics of attack surface help improve the scheme's effectiveness.

5.2. Hardware Testbed-Based. Hardware testbed-based evaluation helps verify the performance of MTD-based techniques under more realistic system environments, using an actual testbed within a lab setting. However, such evaluations do not always scale well for cloud-scale systems.

Regardless, many MTD-based DoS defense works use small or mid-sized hardware testbeds to study and assess the performance of their proposed MTD-based techniques. Among these, [105, 110–116, 127, 130] are notable.

5.2.1. Security Metrics. Among these works that evaluate security metrics, works such as [105, 110, 111, 116] use attack probability as their measure for security. In [105], the authors used probabilistic models to measure the effectiveness of the proposed IP-multiplexing-based network shuffling techniques in terms of attack probability and defense cost. Their experimental setup was created on their lab Intel Xeon server where the results indicate that the proposed framework and algorithm can provide the same security results as known methods but are more cost-effective, while the authors in [110] used probabilistic models to verify the effectiveness of a port hopping-based MTD technique against reconnaissance attacks. In [111], the authors considered resource availability as an important metric for analyzing impact of MTD countermeasure. The system reconfiguration rate is modeled as a function of system resources using continuous-time Markov chain (CTMC). The analysis of the effect of reconfiguration on the availability is considered for fine-tuning MTD decision. Works such as [116] use a probabilistic model by building a stochastic model to describe an integrated defense system consisting of MTD-based maneuvering, deception, and an IDS. They analyzed the performance of the integrated defense system compared to a system with various combinations of defense mechanisms. The hardware testbed was set up using Mininet and Floodlight controller [143]. The evaluation shows that the proposed scheme can minimize the impact of attacks similar to original RRM, while it brings significantly less overhead.

Among the works with attack graph based security evaluation approaches, the authors in [111] argued that an attack graph can be easily visualized and can help the network administrators identify the vulnerabilities of the network and chose appropriate defensive strategies such as MTD. In [112], the authors proposed a SDN-based route mutation technique to deal with DDoS attacks that is validated via a Mininet [141] implementation with a Floodlight SDN controller [143]. Further, they defined a route mutation MTD technique for the ISP network context through NFV and virtual shadow network aiming to thwart possible DDoS attack. They demonstrated that their route mutation method makes it difficult for the attackers to perform attack reconnaissance phase and obtain network topology information.

Among risk evaluation approaches, works such as [113, 114] conducted a substantial analysis to evaluate the effectiveness of two MTD techniques that combine address shuffling and resource diversity. They considered three key metrics for security evaluation, viz., risk (risk), attack cost (AC), and return on attack (RoA). They demonstrated that MTD decreases risk and RoA while increasing AC. The authors also showed that combining shuffling and diversity can optimally meet these multiple objectives, whereas a single solution with either shuffling or diversity cannot. In

[113], the authors used an anti-coordination game to capture the interplay of choice, diversity, and scalability of risk in SDN-based MTD. This study evaluates a scenario where one node in a network is compromised while the others use a game-theoretic approach to decide whether to switch or not. They extended their work in [114] by investigating eight security metrics to evaluate the effectiveness of combined shuffling and diversity. Another work [129] considered a statistical approach to evaluate the likelihood of a successful attack as risk. The authors proposed an approach to determine the minimum effort required from a system to detect stealthy botnets. The entropy was measured to determine how close an adversary is to the detection point, where high entropy indicates the attacker is far from the detector in terms of network distance. The authors used physical servers to create a SDN network consisting of VMs, SDN switches, and OpenDaylight controller [10]. The evaluation results indicated that the proposed mechanism can mitigate DDoS attacks while outperforming other existing algorithms in terms of required CPU overhead.

5.2.2. Usability Metrics. Among the works that evaluate usability on a hardware testbed, [112, 115, 129] measure usability in terms of cost. The authors in [112] presented a cost-effective MTD solution against DDoS and covert channel attacks. Through MTD adaptation, their work aims to answer two main questions. (1) What is the adaptation cost? (2) What is the cost incurred by a defender if an attacker succeeds in exploiting a particular vulnerability? The adaptation cost includes any cost related to purchasing required software or hardware helping in the adaptation process. Their solution does not rely on IDS-generated alerts while making the adaptation. In [115], the authors utilized the change-point analysis method for MTD cost-benefit analysis for a multilayer network resource graph. The proposed method analyzes mission productivity and attack success productivity on dynamic network address translation (DNAT). The evaluation results show reduced attack success probability using DNAT over a network under observation. The path enumeration mechanism used in this research work can, however, suffer from scalability challenges because of frequent path probability calculation and update operations. In [129], the authors developed MASON, a periodic VM migration scheme based on the balance between the level of security obtained and the cost incurred upon the migration of VMs. The experiments are set up on a real science DMZ testbed that consists of VMs, OpenFlow switches, and OpenDaylight controller. The evaluation results show that MASON can effectively thwart DDoS attacks. The results also indicate that situational awareness based on static vulnerability information and dynamic threat events should be used for taking MTD decisions, especially for large-scale cloud networks.

Among the works that measure usability in terms of performance, the authors in [115, 127] conducted a statistical analysis of static vs. dynamic attacks against different MTD strategies: uniform, random, diversity-based, evolution-based, and optimal. Experimental results on

performance vs. adaptability show that diversity-based MTD is the optimal strategy against most attack scenarios. The authors in [130] modeled performance parameters such as availability, downtime, and downtime cost using a continuous-time Markov chain model. The experimental results show that cost-effective VM migration can be performed in a SDN-based network with limited impact on network performance. The research work utilizes normalized CVSS score as a key metric for initiating VM migration.

5.3. Cloud Testbed-Based. Cloud testbed-based evaluations are probably the most widely used validation methods due to the wide availability of community cloud testbeds such as AWS [2], GENI [5], CloudLab [6], DeterLab [144], Chameleon Cloud [145], and PlanetLab [146]. Cloud testbeds provide high level of programmability to implement diverse types of attacks within a controlled environment as well as the ability to implement restriction-free and easily parameterized defense strategies at cloud scale. At the same time, cloud scale implementation allows researchers to gain meaningful insights from “in-the-wild” experiments before implementation in a real system. Finally, results obtained through cloud testbeds are easily reproducible and thus are widely accepted. Therefore, a wide range of works [117, 125, 128, 131–134] in MTD-based DoS defense for cloud environments choose cloud testbed-based evaluations to demonstrate system effectiveness.

5.3.1. Security Metrics. Among works evaluating attack probability, the authors in [125] modeled the security of a configuration as inversely proportional to the probability with which an adversary can come up with a new attack given the attacks it performed in the earlier time steps. Their evaluation is conducted in AWS-based cloud testbed for different case studies with distributed probing to demonstrate the success of the attackers in identifying the number of VMs. The authors in [128] proposed a game-theoretic strategy as a deception technique for MTD to prevent remote OS fingerprinting attacks. They set up their experiments on AWS testbed to show that their proposed technique can significantly decrease the fingerprinting attack success probability while the overall usability of the system is preserved without performance degradation. In [132], the testbed was developed in GENI with VMs, OpenFlow switches, and SDN controller. The authors used response time and average packet dropped to evaluate their reactive scheme. Besides, the attack success rate was used to measure the performance of the proactive scheme. The evaluation indicated that proactive scheme successfully performs migrations that protect the target applications from DDoS attacks with a very low attack success rate, while reactive scheme can effectively mitigate DDoS attacks. Results also show that the false reality scheme successfully tricks an attacker with a false sense of success without substantially increasing the overall CSP cost.

For attack graph-based security evaluation, Bayesian attack graphs have been used by authors in [133, 134] for defending the network against vulnerability exploitation

attempts. In [134], the defender's problem was formulated as a partially observable Markov decision process and the optimal defense policy for selecting countermeasures was identified as a solution. The experimental environment was set up in their private cloud testbed, and the results showed that the proposed mechanism can save 80% of legitimate clients for a DDoS attack of 100K bots. The authors in [133] showed that the security analysis of a large-scale cloud network in real time is a challenging problem. Here, attack graphs help in identification of possible attack scenarios that can lead to exploitation of vulnerabilities in the cloud network. The testbed was built in GENI cloud that simulates a campus network with campus private cloud. The experiment results showed that the utility of VM live migration-based MTD strategy was successful in minimizing the attack impact and future attack success probability.

5.3.2. Usability Metrics. The cost and effectiveness evaluation of reactive and proactive network defense strategies was conducted by works such as [128] using measurement of effectiveness metrics. This work considers hop delay for different attack success rates and static defense policies. They showed that an attacker's productivity, i.e., how quickly attacker can perform adversarial tasks, increases against static defense, whereas attacker's confidentiality, i.e., ability to remain undetected, is the same for both the static and the dynamic defense cases.

For evaluating performance, the authors in [125] tried to solve a multifaceted problem where the MTD tries to obfuscate the network topology to an attacker and, at the same time, ensures that it does not negatively impact a defender's ability to debug network issues. This is done by leveraging the knowledge asymmetry about the network topology that a defender and an attacker has. The authors in [134] analyzed the performance impact of placing IDS at all possible enforcement points in a cloud network. It is noteworthy that the placement of more than 15 detection agents in their simulated network fails to provide any additional intrusion detection benefit, whereas the network throughput decreases drastically from 16 Gbps in the case of a single detection agent to 6 Gbps when 15 detection agents are placed.

6. Research Challenges and Future Directions

Here, we discuss the open challenges and future directions in MTD for cloud DoS defense research domain.

6.1. More Fine-Grained Research on Proactive MTD. Effective proactive or preventive MTD strategies can be designed in order to evade DoS attacks before they hit their target. With the emergence of programmable technology such as SDN, network can be designed where effective anomaly detection can trigger MTD if and when an impending DoS attack is suspected. However, any false positive detection would cause considerable resource wastage from MTD-related resource maneuvering which can add up quickly especially if the system is resource constrained. At the same time, too infrequent maneuvering

can leave the resources vulnerable to attacks and thus eventual service performance degradation. Thus, the fundamental questions to address for effective and efficient MTD design for cloud infrastructure are as follows. (i) What is the optimal frequency of proactive MTD-related resource maneuvering that protects the system without consuming excessive cloud resources? (ii) How to ensure that such frequent proactive maneuvering does not affect the performance of the cloud-hosted services?

6.2. Strong Coupling between MTD and Intrusion Detection/Prevention Systems (IDSs/IPSs). Most of the state-of-the-art IDS/IPS research studies do not include a recovery plan, especially in SDN-based systems where DoS attacks can be more sophisticated to detect and prevent (e.g., table-miss attack). There are some siloed IDS/IPS works in cloud that solely detect and prevent DoS attacks based on artificial intelligence/machine learning (AI/ML) [147]. However, very few of these provide strategies where the effects of the attack can be minimized and/or cloud assets are moved to safety. Thus, there is a need for holistic approaches of IDS/IPS and MTD-based recovery/evasion techniques. With the help of a customized IDS/IPS, more intelligent MTD strategies can be designed for early detection of sophisticated attack signatures and consequent early evasion and recovery.

6.3. Lack of AI/ML for MTD. In recent times, AI/ML has evolved as powerful tools towards defending against cyberattacks and privacy preservation. Although most existing MTD strategies assume some stochastic attack behavior, this might not always be true. Thus, AI/ML integration with MTD is an obvious extension where more effective evasion and recovery strategies can be designed based on robust learning and without preconceived assumptions. Although AI/ML has been used for IDS/IPS in works such as [147], their integration with MTD strategies is still lacking. Therefore, more research is needed towards AI/ML-driven MTD strategy design. However, research is needed to tackle the typical AI/ML challenges such as training latency and requirement of huge datasets in order for such integration to be effective.

6.4. DoS Vulnerabilities for Broader SD Ecosystem. As mentioned before, the research space of DoS attacks and defense on cyber systems is not new, and in recent times, more focus is given on DoS vulnerabilities in cloud systems and cloud-hosted services. Consequently, exploration of DoS vulnerabilities in SDN systems has also gained momentum as most cloud systems are SDN enabled. However, software-defined ecosystems extend far beyond SDN-based cloud data centers. Some examples of such frontier research spaces include SD-RAN (software-defined radio access network), SD-WAN (software-defined wide area network), SDX (software-defined Internet exchange points), and SDx (software-defined everything environments) to name a few. Being software-defined, such ecosystems suffer from the same vulnerabilities as SDN from DoS attacks among many

others which are specific to the use cases supported by these ecosystems. Thus, there is a need for more dedicated research on such new frontiers in broader SD ecosystem.

6.5. Lack of Accessible DoS in Cloud Dataset for Researchers. Another pandemic in cyberattack and defense research space is the lack of state-of-the-art datasets available to researchers. Typically, network attack datasets belonging to Internet service providers (ISPs) are shared and curated through facilities such as CAIDA [148], IMPACT [149], and Kaggle [150]. The ISPs are incentivized to share such data. However, that is not quite true for cloud ecosystems as CSPs such as Google, Amazon, and Microsoft are reluctant to share DoS attack datasets in the fear of disclosing their secret sauce in terms of network design and proprietary protocols. This is quite detrimental to the entire cloud security research community. Thus, there is a need to incentivize the CSP community (maybe through brokering by the federal agencies) in order to ensure more collaboration and co-operation between industry and academia around access to datasets.

6.6. MTD for Private and Community Cloud Systems. Finally, we argue that most of the MTD-based defense strategies consider SDN-based public cloud ecosystem under the control of corporations such as Amazon and Google. However, very little MTD-based defense research is being done for private and community cloud platforms such as institutional cloud facilities and high-performance computing centers (HPCs). These private and community clouds in many cases lack the state-of-the-art cyber defense tools and facilities as (a) they are less visible to the rest of the Internet and consequently are relatively less attractive or lucrative targets of sophisticated cyberattacks and (b) they have overall operating budget constraints and lack resource redundancy. Thus, many of such private and community clouds are ill-equipped to handle sophisticated DoS attacks if and when they occur [133]. Therefore, there is a need to explore more cost-effective, simpler to implement, and more proactive MTD-based defense strategies that do not rely on resource redundancy.

7. Discussion and Conclusions

In this survey, we extensively studied recent notable works that explore how MTD can protect cloud infrastructures. We offered a novel categorization of MTD approaches based on maneuvering techniques such as IP shuffling, live migration, and proxy systems. We classified DoS attacks based on their properties, e.g., volumetric attacks, protocol attacks, and application attacks. Besides, we studied non-MTD methods and DoS defense approaches for non-cloud-like environments. Unlike existing surveys, we extensively discussed the role of SDN in implementing effective MTD-based techniques. We also examined various evaluation methodologies for MTD-based mitigation techniques and provided our perspectives on open challenges and future directions in this space. The discussions of this survey will aid cyber security

domain scientists—beginners and experts alike, cloud service providers, and network administrators in comprehensively understanding the state of the art in this space and exploring the open challenges.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] P. Mell and T. Grance, "NIST special publication 800-145: The NIST definition of cloud computing," *National Institute of Standards and Technology (NIST) - US Department of Commerce*, 2011.
- [2] AWS, "Amazon web services (AWS)," 2022, <https://aws.amazon.com/>.
- [3] Azure, "Microsoft azure," 2022, <https://azure.microsoft.com/>.
- [4] Google, "Google cloud," 2007, <https://cloud.google.com/>.
- [5] C. Elliott, "GENI (global environment for network innovations)," in *Proceedings of the 2008 33rd IEEE Conference on Local Computer Networks (LCN)*, Montreal, QC, Canada, October 2008.
- [6] CloudLab, "CloudLab," 2020, <https://cloudlab.us/>.
- [7] Open Networking Foundation (ONF), "Software-Defined networking (SDN) definition," 2011, <https://opennetworking.org/sdn-definition/>.
- [8] T. R.-535 ONF, "ONF SDN Evolution," 2016.
- [9] N. McKeown, T. Anderson, H. Balakrishnan et al., "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 38, 2008.
- [10] OpenDaylight, "OpenDaylight (O. D. L)," 2021, <https://www.opendaylight.org/>.
- [11] OpenStack, "OpenStack," 2022, <https://www.openstack.org/>.
- [12] K. Yap, T. Huang, B. Dodson, M. S. Lam, and N. McKeown, "Towards Software-Friendly Networks," in *Proceedings of the 1st ACM SIGCOMM Asia-Pacific Workshop on Systems, ApSys*, New Delhi, India, August 2010.
- [13] R. Fielding, "Architectural styles and the design of network-based software architectures," Doctoral Dissertation, University of California, California, Irvine, 2000.
- [14] US Computer Emergency Readiness Team (CERT), "Recommended Practice: improving industrial control system cybersecurity with defense-in-depth strategies," US Department of Homeland Security (DHS), Washington, D.C., USA, Cybersecurity And Infrastructure Security Agency (CISA), 2016.
- [15] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The Science DMZ: a network design pattern for data-intensive science," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Denver, CO, USA, November 2013.
- [16] I. Monga, E. Pouyoul, and C. Guok, "Software-defined networking for big data science - Architectural Models from Campus to the WAN," in *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, Florence, South Carolina, November 2012.
- [17] United States Computer Emergency Readiness Team (Cert), "CERT Security Tip ST04-015: understanding denial-of-service attacks," US Department of Homeland Security (DHS), Washington, D.C., USA, Cybersecurity And Infrastructure Security Agency (CISA), 2019.

- [18] M. Donner, "Phagocytes in cyberspace," *IEEE Symposium on Security and Privacy (S&P)*, vol. 8, 2010.
- [19] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and denial of service attacks: characterization and implications for Cdns and websites," in *Proceedings of the ACM 11th International Conference on World Wide Web (WWW)*, Honolulu, Hawaii, 2002.
- [20] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds," in *Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, Massachusetts, USA, May 2005.
- [21] Global 500. 24/7/365, "Summer 2018 State of the Internet - Security Report," Akamai Technologies, Cambridge, Massachusetts, Global 500. 24/7/365, 2018.
- [22] IAFF 258 - GWU, "State of the Internet - Security: A Year in Review," Akamai Technologies, Cambridge, Massachusetts, IAFF 258 - GWU, 2019.
- [23] Cloudflare, "Famous DDoS attacks: The Largest DDoS Attacks of All Time," 2022, <https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/>.
- [24] M. Pinho, "AWS shield threat landscape report is now available," 2020, <https://aws.amazon.com/blogs/security/aws-shield-threat-landscape-report-now-available/>.
- [25] GitHub, "GitHub," 2022, <https://github.com/>.
- [26] S. Kottler, "February 28th DDoS Incident Report," 2018, <https://github.blog/2018-03-01-ddos-incident-report/>.
- [27] Oracle DNS, "Oracle DNS Dyn," 2019, <https://oci.dyn.com/>.
- [28] Oracle DNS, "Update Regarding DDoS Event against Dyn Managed DNS on October 21, 2016," 2016, <https://www.dynstatus.com/incidents/5r9mppc1kb77>.
- [29] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can We Beat DDoS Attacks in Clouds?" *IEEE Transactions On Parallel And Distributed Systems (TPDS)*, vol. 25, 2014.
- [30] Darpa-Baa-15-56, *Extreme DDoS Defense (XD3) - Amendment 2*, Defense Advanced Research Projects Agency (DARPA), 2015.
- [31] L. M. Marvel, S. Brown, I. Neamtiu, R. Harang, D. Harman, and B. Henz, "A Framework to Evaluate Cyber Agility," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Tampa, FL, USA, October 2015.
- [32] A. Chowdhary, S. Pisharody, and D. Huang, "SDN Based Scalable MTD Solution in Cloud Network," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Moving Target Defense (MTD)*, New York, NY, USA, October 2016.
- [33] G. Cai, B. Wang, W. Hu, and T. Wang, "Moving Target Defense: State of the Art and Characteristics," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, 2016.
- [34] J. Zheng and A. S. Namin, "A Survey on the moving target defense strategies: an architectural perspective," *Springer Journal of Computer Science and Technology*, vol. 34, 2019.
- [35] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A Survey of moving target defenses for network Security," *IEEE Communications Surveys & Tutorials*, vol. 22, 2020.
- [36] J. H. Cho, D. P. Sharma, H. Alavizadeh et al., "Toward Proactive, adaptive defense: a survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, 2020.
- [37] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, 2016.
- [38] N. Agrawal and S. Tapaswi, "Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, 2019.
- [39] O. Yurekten and M. Demirci, "SDN-based cyber defense: a survey," *Elsevier Future Generation Computer Systems (FGCS)*, vol. 115, 2020.
- [40] E. Baize, "Developing secure products in the age of advanced persistent threats," *IEEE Symposium on Security and Privacy (S&P)*, vol. 10, 2012.
- [41] Kaspersky, "What is an advanced persistent threat (APT)?," 2022, <https://www.kaspersky.com/resource-center/definitions/advanced-persistent-threats>.
- [42] Cloudflare, "What is a DDoS Attack?," 2022, <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>.
- [43] E. Osterweil, A. Stavrou, and L. Zhang, "20 Years of DDoS: A Call to Action," 2019, <https://arxiv.org/abs/1904.02739#:~:text=Botnet%20Distributed%20Denial%20of%20Service,trending%20in%20favor%20of%20attackers>.
- [44] MIT Technology Review, "The first DDoS attack was 20 years ago. This is what we've learned since," 2019, <https://www.technologyreview.com/2019/04/18/103186/the-first-ddos-attack-was-20-years-ago-this-is-what-weve-learned-since/>.
- [45] US Computer Emergency Readiness Team (CERT), "CERT Incident Note IN-99-07," US Department of Homeland Security (DHS), Washington, D.C., USA, Cybersecurity And Infrastructure Security Agency (CISA), 1999.
- [46] J. Postel, *RFC 792: Internet Control Message Protocol*, ISOC Request for Comments (RFC), 1981.
- [47] P. Mockapetris, *RFC 1035: Domain Names - Implementation and Specification*, ISOC Request for Comments (RFC), 1987.
- [48] D. Mills, J. Martin, J. Burbank, and W. Kasch, *RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification*, ISOC Request for Comments (RFC), 2010.
- [49] Y. Y. Golland, T. Cai, P. Leach, Y. Gu, and S. Albright, *Internet Draft: Simple Service Discovery Protocol*, ISOC Internet Draft, vol. 1.0, 1999.
- [50] A. Young, *RFC 1798: Connection-Less Lightweight X.500 Directory Access Protocol*, ISOC Request for Comments (RFC), 1995.
- [51] Memcached, "Memcached," 2022, <https://memcached.org/>.
- [52] C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed Denial of Service Attacks," *The Internet Protocol Journal (IPJ)-Cisco Systems*, vol. 7, no. 4, pp. 13–35, 2004.
- [53] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, USA, May 1997.
- [54] US Computer Emergency Readiness Team (CERT), "CERT Advisory CA-1996-21: TCP SYN flooding and IP spoofing attacks," US Department of Homeland Security (DHS), Washington, D.C., USA, Cybersecurity And Infrastructure Security Agency (CISA), 1996.
- [55] J. Postel, *RFC 793: Transmission Control Protocol*, ISOC Request for Comments (RFC), 1981.
- [56] J. Mogul and S. Deering, *RFC 1191: Path MTU Discovery*, ISOC Request for Comments (RFC), 1990.
- [57] Microsoft, "The Default MTU Sizes for different network topologies," 2008, <https://support.microsoft.com/en-us/topic/the-default-mtu-sizes-for-different-network-topologies-b25262c5-d90f-456d-7647-e09192eeef4>.

- [58] J. Postel, *RFC 791: Internet Protocol*, ISOC Request for Comments (RFC), 1981.
- [59] R. van den Berg and P. Dibowitz, "Over-zealous security administrators are breaking the internet," in *Proceedings of the 16th USENIX Conference on System Administration (LISA'02)*, USENIX Association, USA, November 2002.
- [60] R. Fielding, J. Gettys, J. Mogul et al., *RFC 2616: Hypertext Transfer Protocol -*, ISOC Request for Comments (RFC), 1999.
- [61] RSnake and J. Kinsella, "Slowloris DDoS attack," 2022, <http://hackers.org/slowloris/>.
- [62] Cloudflare, "What is a Slowloris DDoS attack?," 2022, <https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/>.
- [63] R.U.D.Y., "What is a R.U.D.Y. attack?," 2021, <https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/>.
- [64] Akamai, "CDN Definition," 2022, <https://www.akamai.com/us/en/cdn/what-is-a-cdn.jsp>.
- [65] NGINX, "What is load balancing?," 2019, <https://www.nginx.com/resources/glossary/load-balancing/>.
- [66] AWS, "What is a DDoS Attack?," 2022, <https://aws.amazon.com/shield/ddos-attack-protection/>.
- [67] Cloudflare, "What is DDoS mitigation?," 2022, <https://www.cloudflare.com/learning/ddos/ddos-mitigation/>.
- [68] D. J. Bernstein, "SYN Cookies," 1996, <https://cr.yp.to/syncookies.html>.
- [69] T. W. Doepfner, P. N. Klein, and A. Koyfman, "Using Router Stamping to Identify the Source of IP Packets," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, New York, NY, USA, November 2000.
- [70] X. Liu, X. Yang, and Y. Xia, "NetFence: preventing internet denial of service from inside Out," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 40, pp. 255–266, 2010.
- [71] J. Ioannidis and S. M. Bellovin, "Implementing pushback: router-based defense against DDoS attacks," *Network and Distributed System Security Symposium (NDSS)*, 2002.
- [72] Cisco, "Cisco Systems, Inc.," 2022, <https://www.cisco.com/>.
- [73] Juniper, "Juniper," 1999, <https://www.juniper.net/us/en.html>.
- [74] Linux, "Linux Foundation," 2022, <https://www.linuxfoundation.org/>.
- [75] FreeBSD, "FreeBSD Project," 1995, <https://www.freebsd.org/>.
- [76] S. Lance, *Honeypots: Tracking Hackers*, Addison-Wesley Professional, Boston, MA, USA, 2002.
- [77] B. McCarty, "Botnets: big and bigger," *IEEE Symposium On Security And Privacy (S&P)*, vol. 1, 2003.
- [78] J. Krupp, M. Backes, and C. Rossow, "Identifying the scan and attack infrastructures behind amplification DDoS attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, New York, NY, USA, October 2016.
- [79] K. Salah, J. M. A. Calero, S. Zeadally, S. Al-Mulla, and M. Alzaabi, "Using cloud computing to implement a security overlay network," *IEEE Symposium on Security and Privacy (S&P)*, vol. 11, 2013.
- [80] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and elastic DDoS defense," in *Proceedings of the 24th USENIX Security Symposium (USENIX Security)*, Washington, D.C., USA, August 2015.
- [81] M. Zhang, G. Li, S. Wang et al., "Poseidon: mitigating volumetric ddos attacks with programmable switches," *Network and Distributed System Security Symposium (NDSS)*, 2020.
- [82] Z. Liu, H. Namkung, G. Nikolaidis et al., "Jaen: A high-performance switch-native approach for detecting and mitigating volumetric ddos attacks with programmable switches," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*, Berkeley, CA, USA, 2021.
- [83] S. Narayana, M. Tahmasbi, J. Rexford, and D. Walker, "Compiling path queries," in *Proceedings of the 13th USENIX Symposium On Networked Systems Design And Implementation (NSDI)*, USA, March 2016.
- [84] M. Elsbagh, D. Fleck, A. Stavrou, M. Kaplan, and T. Bowen, "Practical and accurate runtime application protection against DoS Attacks," *International Symposium On Research In Attacks, Intrusions And Defenses (RAID)*, Springer, Manhattan, NY, USA, 2017.
- [85] H. M. Demoulin, I. Pedisich, N. Vasilakis, V. Liu, B. T. Loo, and L. T. X. Phan, "Detecting Asymmetric Application-Layer Denial-Of-Service Attacks In-Flight with FINELAME," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, Renton, WA, USA, July 2019.
- [86] Y. Gilad, A. Herzberg, M. Sudkovitch, and M. Goberman, "CDN-on-Demand: An Affordable DDoS Defense via Untrusted Clouds," *Network and Distributed System Security Symposium (NDSS)*, 2016.
- [87] S. Ramanathan, J. Mirkovic, M. Yu, and Y. Zhang, "SENSS against Volumetric DDoS Attacks," in *Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC)*, New York, NY, USA, December 2018.
- [88] S. Zheng and X. Yang, "DynaShield: reducing the cost of ddos defense using cloud services," in *Proceedings of the 11th USENIX Workshop On Hot Topics In Cloud Computing (HotCloud)*, Renton, WA, USA, July 2019.
- [89] T. Vissers, T. V. Goethem, W. Joosen, and N. Nikiforakis, "Maneuvering around clouds: bypassing cloud-based security providers," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, New York, NY, USA, October 2015.
- [90] J. Bushart and C. Rossow, "DNS Unchained: Amplified Application-Layer DoS Attacks against DNS Authoritatives," in *Lecture Notes in Computer Science* Springer, Cham, Manhattan, NY, USA, 2018.
- [91] R. Jansen, T. Vaidya, and M. Sherr, "Point Break: a study of bandwidth denial-of-service attacks against tor," in *Proceedings of the 28th USENIX Security Symposium (USENIX Security)*, Berkeley, CA, USA, August 2019.
- [92] D. Kopp, M. Wichtlhuber, I. Poese, J. Santanna, and O. Hohlfeld, "DDoS hide & seek: on the effectiveness of a booter services takedown," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, New York, NY, USA, October 2019.
- [93] Torproject, "Tor," 2006, <https://www.torproject.org/>.
- [94] G. Gu, D. Ott, V. Sekar, and K. Sun, "Programmable System Security in a Software-Defined World – Research Challenges and Opportunities," Technical Report, NSF Workshop On Programmable System Security In a Software Defined World, Arlington, Va, USA, NSF Workshop On Programmable System Security In a Software Defined World, 2018 .
- [95] M. C. Dacier, S. Dietrich, F. Kargl, and H. Konig, "Network attack detection and defense – security challenges and opportunities of software-defined networking," *Dagstuhl Seminar*, vol. 16361, 2016.
- [96] M. C. Dacier, H. Konig, R. Cwalinski, F. Kargl, and S. Dietrich, "Security challenges and opportunities of

- software-defined networking,” *IEEE Symposium on Security and Privacy (S&P)*, vol. 15, 2017.
- [97] M. Zhang, G. Li, L. Xu, J. Bi, G. Gu, and J. Bai, “Control plane reflection attacks in SDNs: new attacks and countermeasures,” in *Lecture Notes in Computer Science* Springer, Cham, Manhattan, NY, USA, 2018.
- [98] M. Zhang, J. Bi, J. Bai, and G. Li, “FloodShield: Securing the SDN Infrastructure against Denial-Of-Service Attacks,” in *Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, NY, USA, August 2018.
- [99] H. Wang, L. Xu, and G. Gu, “FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks,” in *Proceedings of the 45th Annual IEEE/IFIP International Conference On Dependable Systems And Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.
- [100] G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, “Flood-Defender: Protecting Data and Control Plane Resources under SDN-Aimed DoS Attacks,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, Atlanta, GA, USA, May 2017.
- [101] G. Shang, P. Zhe, X. Bin, H. Aiqun, S. Yubo, and R. Kui, “Detection and Mitigation of DoS Attacks in Software Defined Networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 28, 2020.
- [102] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut, “Analysis of Network Address Shuffling as a Moving Target Defense,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, June 2014.
- [103] H. Alavizadeh, J. Jang-Jaccard, and D. S. Kim, “Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing,” in *Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications/IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, NY, USA, August 2018.
- [104] H. Alavizadeh, J. B. Hong, J. Jang-Jaccard, and D. S. Kim, “Comprehensive Security Assessment of Combined MTD Techniques for the Cloud,” in *Proceedings of the ACM Conference On Computer And Communications Security (CCS) - Workshop On Moving Target Defense (MTD)*, New York, NY, USA, October 2018.
- [105] H. Wang, F. Li, and S. Chen, “Towards Cost-Effective Moving Target Defense against DDoS and Covert Channel Attacks,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Moving Target Defense (MTD)*, Vienna, Austria, October 2016.
- [106] A. Clark, K. Sun, and R. Poovendran, “Effectiveness of IP Address Randomization in Decoy-Based Moving Target Defense,” in *Proceedings of the IEEE 52nd Conference on Decision and Control (CDC)*, Firenze, Italy, December 2013.
- [107] A. Clark, K. Sun, L. Bushnell, and R. Poovendran, “A Game-Theoretic Approach to IP Address Randomization in Decoy-Based Cyber Defense,” *Lecture Notes in Computer Science*, Springer International Conference on Decision and Game Theory for Security, Manhattan, NY, USA, 2015.
- [108] F. Nizzi, T. Pecorella, F. Esposito, L. Pierucci, and R. Fantacci, “IoT security via address shuffling: the easy way,” *IEEE Internet of Things Journal (IoT-J)*, vol. 6, 2019.
- [109] S. Yao, Z. Li, J. Guan, and Y. Liu, “Stochastic cost minimization mechanism based on identifier network for iot security,” *IEEE Internet of Things Journal (IoT-J)*, vol. 7, 2020.
- [110] P. Kampanakis, H. Perros, and T. Beyene, “SDN-based solutions for moving target defense network protection,” in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Sydney, NSW, Australia, June 2014.
- [111] J. Steinberger, B. Kuhnert, C. Dietz et al., “DDoS Defense Using MTD and SDN,” in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Taipei, Taiwan, April 2018.
- [112] Y. Zhou, G. Cheng, S. Jiang, Y. Hu, Y. Zhao, and Z. Chen, “A cost-effective shuffling method against ddos attacks using moving target defense,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Moving Target Defense (MTD)*, London, UK, November 2019.
- [113] J. H. Jafarian, E. Al-Shaer, and Q. Duan, “Openflow random host mutation: transparent moving target defense using software defined networking,” in *Proceedings of the ACM SIGCOMM - Workshop on Hot Topics in Software Defined Networks (HotSDN)*, Helsinki, Finland, August 2012.
- [114] J. H. Jafarian, E. Al-Shaer, and Q. Duan, “An effective address mutation approach for disrupting reconnaissance attacks,” *IEEE Transactions on Information Forensics and Security (TIFS)*, 2015.
- [115] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, “MTD analysis and evaluation framework in software defined network (MASON),” in *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFV Sec)*, Tempe, AZ, USA, March 2018.
- [116] A. Aydeger, M. H. Manshaei, M. A. Rahman, and K. Akkaya, “Strategic defense against stealthy link flooding attacks: a signaling game approach,” *IEEE Transactions on Network Science and Engineering*, vol. 8, 2021.
- [117] M. Nguyen, A. Pal, and S. Debroy, “Whack-a-mole: software-defined networking driven multilevel ddos defense for cloud environments,” in *Proceedings of the IEEE 43rd Conference on Local Computer Networks (LCN)*, Chicago, IL, USA, October 2018.
- [118] J. B. Hong and D. S. Kim, “Towards scalable security analysis using multi-layered security models,” *Elsevier Journal of Network and Computer Applications*, vol. 75, 2016.
- [119] ONOS, “Open network operating system,” 2022, <https://opennetworking.org/onos/>.
- [120] J. H. Jafarian, E. Al-Shaer, and Q. Duan, *Formal Approach for Route Agility against Persistent Attackers*, Springer European Symposium on Research in Computer Security, Berlin Heidelberg, 2013.
- [121] C. Xu, T. Zhang, X. Kuang, Z. Zhou, and S. Yu, “Context-aware adaptive route mutation scheme: a reinforcement learning Approach,” *IEEE Internet of Things Journal (IoT-J)*, vol. 8, 2021.
- [122] Q. Jia, K. Sun, and A. Stavrou, “MOTAG: Moving target defense against internet denial of service attacks,” in *Proceedings of the IEEE 22nd International Conference on Computer Communications and Networks (ICCCN)*, Nassau, Bahamas, July 2013.
- [123] H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, and A. Stavrou, “A moving target ddos defense mechanism,” *Elsevier Computer Communications*, vol. 46, 2014.
- [124] P. Wood, C. Gutierrez, and S. Bagchi, “Denial of service elusion (DoSE): keeping clients connected for less,” in *Proceedings of the IEEE 34th Symposium On Reliable Distributed Systems (SRDS)*, Montreal, QC, Canada, September 2015.

- [125] D. Fleck, A. Stavrou, G. Kesidis, N. Nasiriani, Y. Shan, and T. Konstantopoulos, *Moving-target Defense against Botnet Reconnaissance and an Adversarial Coupon-Collection Model*, IEEE Conference on Dependable and Secure Computing (DSC), 2018.
- [126] G. Kesidis, Y. Shan, D. Fleck, A. Stavrou, and T. Konstantopoulos, "An adversarial coupon-collector model of asynchronous moving-target defense against botnet reconnaissance," in *Proceedings of the IEEE 13th International Conference on Malicious and Unwanted Software (MALWARE)*, Kaohsiung, Taiwan, December 2018.
- [127] N. Bandi, H. Tajbakhsh, and M. Analoui, "FastMove: fast IP switching moving target defense to mitigate DDOS attacks," in *Proceedings of the IEEE Conference on Dependable and Secure Computing (DSC)*, Aizuwakamatsu, Fukushima, Japan, January 2021.
- [128] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright, *A Moving Target Defense Approach to Mitigate DDoS Attacks against Proxy-Based Architectures*, IEEE Conference on Communications and Network Security (CNS), 2016.
- [129] M. Wright, S. Venkatesan, M. Albanese, and M. P. Wellman, "Moving target defense against DDOS attacks: an empirical game-theoretic analysis," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Moving Target Defense (MTD)*, Vienna, Austria, October 2016.
- [130] A. Aydeger, N. Saputro, and K. Akkaya, "Utilizing NFV for effective moving target defense against link flooding reconnaissance attacks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA, USA, October 2018.
- [131] S. Debroy, P. Calyam, M. Nguyen, A. Stage, and V. Georgiev, "Frequency-minimal moving target defense using software-defined networking," in *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)*, Kauai, HI, USA, February 2016.
- [132] S. Debroy, P. Calyam, M. Nguyen et al., "Frequency-minimal utility-maximal moving target defense against DDOS in SDN-based systems," *IEEE Transactions on Network and Service Management (TNSM)*, vol. 17, 2020.
- [133] M. Nguyen, P. Samanta, and S. Debroy, "Analyzing moving target defense for resilient campus private cloud," in *Proceedings of the IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, July 2018.
- [134] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch me if you can: a cloud-enabled DDOS defense," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DNS)*, Atlanta, GA, USA, June 2014.
- [135] W. Peng, F. Li, C. T. Huang, and X. Zou, "A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, June 2014.
- [136] S. Venkatesan, M. Albanese, G. Cybenko, and S. Jajodia, "A moving target defense approach to disrupting stealthy botnets," in *Proceedings of the ACM Conference On Computer And Communications Security (CCS) - Workshop On Moving Target Defense (MTD)*, Vienna, Austria, October 2016.
- [137] CVSS, "CVSS specification document," 2021, <https://www.first.org/cvss/specificationdocument>.
- [138] CVE, "CVE," 2021, <https://cve.mitre.org/>.
- [139] F. Gillani, E. Al-Shaer, and Q. Duan, "In-design resilient SDN control plane and elastic forwarding against aggressive DDOS attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Moving Target Defense (MTD)*, Toronto, Canada, January 2018.
- [140] MATLAB, "MATLAB," 2021, <https://www.mathworks.com/products/matlab.html>.
- [141] Mininet, "Mininet," 2022, <http://mininet.org/>.
- [142] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the ACM SIGCOMM - Workshop on Hot Topics in Networks (HotNets)*, Monterey, California, October 2010.
- [143] Floodlight, "Floodlight," 2020, <https://floodlight.atlassian.net/>.
- [144] DeterLab, "DeterLab," 2021, <https://www.isi.deterlab.net/>.
- [145] Chameleon, "Chameleon," 2021, <https://www.chameleoncloud.org/>.
- [146] PlanetLab, "PlanetLab," 2021, <https://planetlab.cs.princeton.edu/>.
- [147] R. S. S. Kumar, A. Wicker, and M. Swann, "Practical machine learning for cloud intrusion detection: challenges and the way forward," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) - Workshop on Artificial Intelligence and Security (AISec)*, Sydney, NSW, Australia, June 2017.
- [148] CAIDA, "CAIDA," 2021, <https://www.caida.org>.
- [149] IMPACT, "MPACT," 2022, <https://www.impactcybertrust.org>.
- [150] Kaggle, "Kaggle," 2020, <https://www.kaggle.com>.