Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

SDN-based cyber defense: A survey

Ozgur Yurekten [a,b], Mehmet Demirci [a,*]

[a] Department of Computer Engineering, Gazi University, Ankara, Turkey
[b] Cyber Security Institute, TUBITAK-BILGEM, Ankara, Turkey

ARTICLE INFO

Article history:
Received 24 January 2020
Received in revised form 25 July 2020
Accepted 4 September 2020
Available online 8 September 2020

Keywords:
Cyber threats
CTI
Network security
Cyber defense
Software defined networking
SDN

ABSTRACT

The growth and ubiquity of the Internet have changed the world in numerous ways, one of which is giving rise to the necessity of being vigilant about information security and cyber threats. As threat actors have become more sophisticated and new threats are emerging constantly, meeting information security objectives requires taking advantage of the latest technologies and tools. This paper focuses on a popular technology that can improve the way security is achieved: software-defined networking (SDN). Thanks to its flexibility, cost efficiency, and suitability for incremental deployment, SDN provides a practical means of developing effective security solutions. Through an extensive survey of the literature, we develop a taxonomy for SDN-based solutions to common attack types, identify the security primitives utilized in these studies, and categorize proposals by cyber threat category. Furthermore, we present a quantitative evaluation of the reviewed studies according to threat category, defense type, strategy, techniques, and deployment details. Finally, we discuss various challenges and potential research questions to be investigated in this area.

1. Introduction

Modern technologies enabling constant connectivity have shaped the current communication patterns, which in turn lead to increasing demands from the underlying technologies. Internet users expect high connection speed, availability, accessibility, mobility support, and foolproof security from their providers. However, popular technologies such as cloud computing, smart devices, and Internet of things (IoT) have caused new security risks as much as they have enhanced our lives. New threats and attack vectors are emerging with every new communication technology, platform, and tool. In the first quarter of 2019, more than 65 million new malware instances were detected and total malware count has reached nearly 1 billion [1]. Furthermore, more than 46 000 variations of crypto-ransomware were detected and 22 new crypto-ransomware families were discovered in 2019 [2]. In addition to malware, other threats like distributed denial of service (DDoS), sniffing/eavesdropping, web application attacks, and advanced persistent threats (APTs) are also increasing. For example, the Mirai botnet consisting of IoT devices was identified in August 2016 [3] and has grown considerably since then.

In traditional networks, there are different security solutions (e.g., firewall, unified threat management, deep packet inspection, intrusion prevention system) to prevent, detect, and mitigate cyber threats. Many of these solutions are offered as specialized hardware appliances, and their deployment locations are carefully determined. Such security solutions have a constraint that their locations and functions are fixed and cannot be changed dynamically. Besides, certain defensive actions such as traffic redirection, honeypot deployment, and network separation are costly and unstable in such networks. Management and configuration of networks are highly complex, challenging, and time consuming tasks [4]. As an emerging paradigm, software-defined networking (SDN) simplifies and improves network management by providing highly flexible networks based on the principle of separating control and data planes [5]. SDN offers network programmability through protocols such as OpenFlow [6], reduces the need for custom network devices, and enables implementing new network algorithms and protocols at the control layer without changing network devices [5]. For these reasons, it provides an ideal platform to develop and test new security solutions.

A growing interest exists for preventive response and proactive cyber defense approaches against cyber threats such as zero day attacks, DDoS, malware, etc. [7]. Manually identifying, analyzing, and then preventing and defending against all cyber threat types is highly difficult. For this reason, sharing cyber threat information has great importance for organizations and various cyber threat intelligence (CTI) data formats are defined such as STIX (Structured Threat Information eXpression) [8], IODEF (The Incident Object Description Exchange Format) [9], OpenIoC (Open Indicators of Compromise) [10] to enable sharing CTI. CTI data define threat indicators collected from multiple sources and may

[*] Corresponding author.
E-mail addresses: ozgur.yurekten@tubitak.gov.tr (O. Yurekten), mdemirci@gazi.edu.tr (M. Demirci).

https://doi.org/10.1016/j.future.2020.09.006
0167-739X/© 2020 Elsevier B.V. All rights reserved.

explain courses of action to deal with these threats [7]. Moreover, CTI data can be used for managing risks to evaluate cyber threats, simulating cyber attacks, and taking precautions to prevent them.

Research works considering SDN and cyber security together have been roughly clustered under four categories: automated derivation of secure SDN configurations, secure operations in SDN environments, SDN-based security, and secure architectures for SDN [11]. Among these, attacks on SDN and SDN-based defense actions against attacks are the most common research topics [12]. The scope of this survey paper is the latter: the use of SDN technology to strengthen cyber defense. To this end, we have conducted a systematic review of the network security literature with a focus on cyber threat categories, which can be defined in and extracted from CTI data, and analyzed the SDN-based defense methods against each type of threat. New attacks exclusively targeting SDN infrastructures and defense methods specifically against these attacks are outside the scope of the paper. In addition, while SDN provides improved attack detection capability thanks to the global view of the centralized controller, the cyber security benefits of SDN can only be fully realized by solutions taking advantage of its other important properties which facilitate and strengthen attack prevention and response, such as the ease of making changes to routing and resource allocation, the ability to filter packets or modify packet headers using flow rules on switches, etc. Thus, we seek studies proposing attack prevention and/or mitigation techniques using the features of SDN. Studies that present approaches for detection along with prevention and/or mitigation are included in our reviews, however, studies proposing solely detection methods for cyber attacks are excluded. In short, we present an extensive survey of SDN-based mitigation and prevention solutions against a variety of common attack types in this paper.

In the literature, several surveys [13–20] have been conducted on SDN-based network security. These surveys review studies on both SDN-based network security and the security of the SDN architecture, so they would be helpful for researchers who are looking for both of these topics. However, each survey categorizes and reviews SDN-based security solutions using a different taxonomy, as shown in Table 1. The main categories of these taxonomies are generally solution-oriented or security requirement-oriented. Only the study by Swami et al. [21] includes a threat-oriented taxonomy and reviews SDN-based defense solutions against DDoS attacks according to three threat categories: flooding, amplification and application-layer attacks. On the other hand, our study presents more threat types covering a much wider spectrum of attacks, which are listed in the last row of Table 1, and reviews relevant studies proposing mitigation and/or prevention methods against these cyber threats. Therefore, the distinguishing features and the novelty of the current study are twofold: It classifies existing studies by cyber threat category and highlights their similarities and differences in terms of defense objectives, methods, and deployment details. Furthermore, it breaks down the approach in each study into its building blocks and identifies SDN-based network defense primitives commonly employed to deal with each type of cyber threat. As a result, researchers and security professionals reading this paper can easily review and contrast different approaches to a certain type of threat at the granularity they desire.

The rest of the paper is organized as follows: Section 2 provides a brief overview of CTI and SDN. Section 3 defines the cyber threat categorization used in this paper by discussing different taxonomies in the literature. SDN-based cyber defense solutions are categorized by type of attack and analyzed in Section 4. In Section 5, an overall taxonomy and various statistics regarding the reviewed studies are presented, and SDN-based defense primitives are evaluated based on how they are utilized in these studies. Section 6 highlights the gaps in the current literature, and discusses some challenges as well as potential future research directions. Finally, Section 7 summarizes and concludes the paper.

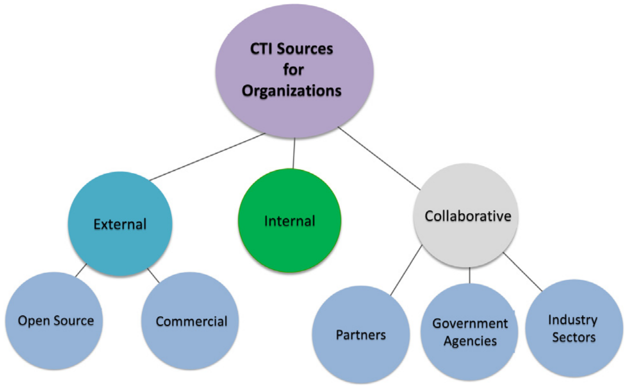


Fig. 1. Cyber threat intelligence sources for organizations.

2. Background

In this study, we first categorize cyber threats by examining the literature and cyber threat intelligence data, and then identify network defense mechanisms used against these threats in software-defined networks. The following subsections provide background on creating, sharing and using cyber threat intelligence (CTI), and summarize the fundamentals of software-defined networking (SDN).

2.1. Cyber threat intelligence

In general, cyber threat intelligence is defined as evidence that is gathered and verified from multiple sources in order to take action to protect specific targets against cyber threats [7,22]. Cyber threat intelligence process consists of five phases [22]:

1. Planning and directing of process,
2. Collecting potentially useful raw data from relevant sources via human intelligence, open source intelligence, signal intelligence, imagery intelligence and measurement and signature,
3. Processing of collected data into a standardized format for detailed analysis,
4. Analyzing processed data to identify threats and to find suitable countermeasures,
5. Disseminating the results in the context of cyber threat information and indicators so that appropriate protective measures can be taken.

Organizations can obtain CTI data from three different sources (Fig. 1). The first source of CTI data is internal capabilities. All security devices and events are collected and analyzed by personnel. But this process is time-consuming, labor-intensive, and potentially error-prone [22]. Alternatively, organizations can receive CTI data from commercial or open community sources. The third way of obtaining CTI data is sharing among partner organizations, such as different government institutions or private companies.

With the increase of cyber threat types and threat vectors, incidents of cyber attacks are also increasing. As a result, security concerns of organizations and governments have reached a critical level. In the traditional network defense approach, cyber threats are collected, identified, grouped by IT experts and countermeasures are taken to neutralize the identified threats. Nowadays, using up-to-date information about the intentions and capabilities of cyber attackers has become a necessity to be able to use proactive cyber defense approaches [7]. New approaches

Table 1
Categories of SDN-based security solutions as defined in related surveys.

Survey	Review taxonomy	Review categories
Alsmadi and Xu [13]	Security controls	Firewalls, access control, IDS/IPS, policy management, monitoring and auditing, mobile security control, Wi-Fi networks, privacy protection, security controls of BYOD (bring your own device), security control of open labs
Ahmad et al. [14]	Security type	Flow sampling, packet sampling, IPS, firewall, flow-based IDS, network resilience, security monitoring, network monitoring, flow rule verification, configuration verification, controller availability
	Security requirements	Access control, authentication, non-repudiation, data confidentiality, communication security, data integrity, availability, privacy
Scott-Hayward et al. [15]	Security enhancements	Collect–detect–protect, traffic analysis & rule updating, DoS/DDoS protection, middlebox architectures & services, authentication–authorization–accounting, secure–scalable multi tenancy
Shaghaghi et al. [17]	SDN-based security services	Intrusion prevention systems, privacy-enhancing services, security middleboxes, protecting cloud services, secure data offloading, IoT security, other
Rawat and Reddy [18]	SDN security solutions	SDN as IDS/IPS, SDN for anomaly detection, SDN for DDoS attack detection and prevention
Chica et al. [19]	Network security	Threat detection, attack remediation, identity and access management, network state monitoring and analysis
Farris et al. [20]	SDN-based security features	Traffic isolation, security network monitoring through centralized visibility, dynamic flow control, host and routing obfuscation, security network programmability
Swami et al. [21]	DDoS defense by SDN	Statistical/policy based defense, machine learning based defense, application specific defense
Our study	Cyber threat categories	Scanning attacks, spoofing attacks, network-level denial of service (DoS) attacks, sniffing attacks, malware and social engineering attacks, web application attacks

must be developed to organize information about attackers' activities, utilities, malware and other indicators of compromise [10]. For this purpose, specifications for creating and sharing cyber threat intelligence (CTI) data have been defined. Some of the most common specifications are STIX [8], OpenIoC [10], TAXII [23] (Trusted Automated eXchange of Intelligence Information), and CybOX [24] (Cyber Observable eXpression). However, CTI data shared according to these specifications are currently being used in a way that is mainly meaningful to humans [25,26]. Most CTI data do not include machine-actionable course of actions for defined threats, and organizations process CTI data and update their security policies via updating the rules of firewalls, end-point protection systems, intrusion prevention systems (IPS), etc.

2.2. Software-defined networking (SDN)

Traditional networks are complex and can be hard to manage. Configuring networks according to organizational policies and managing changes are difficult processes. Furthermore, control planes and data planes of current networks are vertically integrated [27], which makes it harder to maintain network devices [5]. Software-defined networking is an architecture that enables flexible management of computer networks via separating the data forwarding plane and control plane. Key features of SDN include providing logically centralized management and programmable interfaces with open standards, improved network switch management protocols, the ability to easily create virtualized logical networks, and enabling the use of centralized monitoring modules [15]. SDN is widely used in data centers, backbone networks, enterprise networks and wireless networks because of its flexibility, programmability and easier maintenance [28].

SDN architecture consists of three layers and inter-layer communication interfaces as shown in Fig. 2 [29]. The SDN switch (1) in the data layer forwards incoming traffic (2) according to the rules in its programmable flow tables and collects statistical information about traffic passing through it. If an incoming packet does not match any of the rules in flow tables, the SDN switch takes the default action of forwarding the packet to the SDN controller (3) on the control layer via the southbound interface (4). The SDN controller decides what to do for the packets coming from the SDN switches and defines new flow rules (forward, update, drop, etc.) for the switches. Using the southbound interface, the controller can also collect statistical information from the switches periodically or on demand. There can be multiple SDN controllers on the control layer. In this case, the controllers communicate with each other via the east–west interface (5). Applications (6) in the application layer (network security, traffic monitoring, etc.) can make changes on the network or monitor the network through the controller via the northbound interface (7).

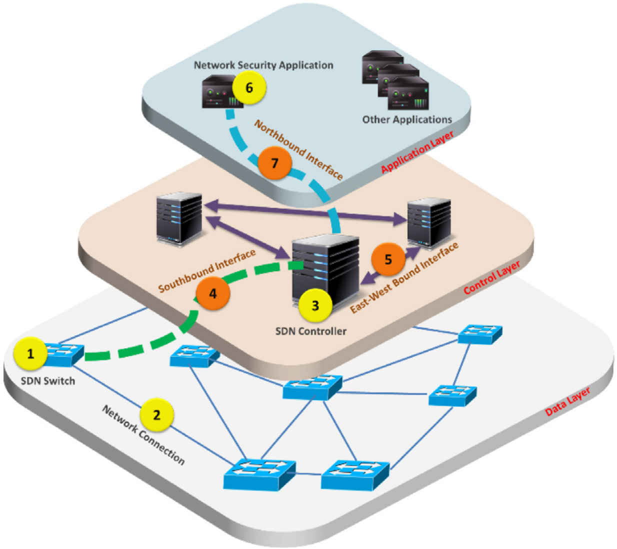


Fig. 2. Software defined network architecture.

SDN is an exciting realm for cyber security because its features such as programmability and centralization of control enable both better defense mechanisms as well as more dangerous attacks [29–32]. In fact, the recent literature contains some devastating attacks [33,34] on SDN systems, and also a large body of SDN-based defense solutions, many of which will be discussed in the coming sections. Beyond more traditional and widely studied domains such as ISP networks and the cloud, SDN has been proposed as a force for improved security in other domains like smart grids [35,36] and industrial networks [37]. This study approaches the SDN-assisted security literature in a threat-oriented manner. We will first present a discussion on categorizing cyber threats, followed by a broad survey on defense proposals utilizing SDN.

3. Threat categories based on CTI

Cyber threat is a malicious action to access and control any computer network, information technology (IT) (e.g., computer, mobile phone) or operational technology (OT) device (e.g., programmable logic controllers (PLC), supervisory control and data acquisition systems (SCADA)).

New cyber threat vectors appear every day, and preventive and defensive actions that tackle these new cyber threat types become more complex. Behaviors and resources that attackers use to carry out their attacks are defined as TTP (tactics, techniques, and procedures) in STIX v1.2.1 [38], In STIX v2.1 specification [8], the same concept is represented in a different way: TTP type in the previous version is divided into attack pattern, malware and tool object types. An attack pattern is an abstraction method for defining how a type of attack is executed [39]. Attack pattern is useful for defining current or new attack vectors. CAPEC (Common Attack Pattern Enumeration and Classification) [40] is a publicly available catalog of common attack patterns classified for describing related attacks and sharing information about threats. Besides, MITRE ATT&CK (MITRE Adversarial Tactics, Techniques, and Common Knowledge) [41] defines common tactics, techniques, and procedures that advanced persistent threats use against enterprise networks.

Network and computer security taxonomies aim to classify cyber security threats. There is not a specific approach for presenting these taxonomies verbally [42]. In the literature, cyber threat categorization and taxonomy studies used different terminologies and reviewed the literature from different perspectives [42–47]. Earlier cyber security taxonomies are based on vulnerabilities rather than attacks and they are explained historically in [43,47] from 2001 back to 1975. Jin et al. [44] review categorization studies published in the years 1976–2008. Canbek et al. [42] summarize 28 security taxonomies about attacks, incidents, malware, threat, and vulnerabilities from 1993 to 2015.

Web attacks, reconnaissance, DoS, service specific attacks and known bad sources have emerged as the most common network attacks [48]. The most observed threats in organizations are phishing (72% of respondents), spyware (50% of respondents), ransomware (49% of respondents) and Trojans (47% of respondents) [49]. The same report notes that phishing has the greatest impact, and DDoS and APT have a greater impact than Trojans or spyware.

Cyber attack categorization studies in the literature are shown Table 2. Analyzing cyber attacks in Table 2, we can make the following inferences:

- Attack taxonomy has been an active research area since 1975 and there is no consensus on cyber attack names.
- The granularity of cyber attack categorizations is not consistent.
- Most attack types are common (i.e., DoS, scanning, spoofing) in studies.
- Categories for the same attacks may change between studies depending on the focus of research.

The studies in Table 2 are not restricted to CTI data usage. In general, cyber attacks can be handled within networks or end devices. This study focuses on cyber threats that can be defined in CTI data, and actions that can be taken on a network. Threat categories used in this study are derived from attacks defined in Table 2, which are categorized and aligned based on network defense according to CTI data models. Threats are defined in CTI data with varying specificity, so it is important to construct a categorization to organize cyber threats. Most studies in the literature focus on specific cyber threats and propose defense solutions against them. In Table 3, we provide a list of common threats encountered in CTI data, and group them under different categories based on the relationships among them to present our investigation in a more structured way. Each category is defined to be comprehensive enough to exist as an attack class on its own, and also specific enough to exhibit meaningful differences from other categories in both attack methodology and proposed defense mechanisms against it. The cyber threat categories in Table 3 include but are not limited to the threats listed in the *Cyber Threats* column of the table. We should also note that certain complex attacks such as APTs and large-scale DDoS attacks may involve threats from multiple categories in the table. Such attacks usually require a multipronged defense mechanism combining different defense approaches. These issues are discussed in more detail in the next section.

4. SDN-based cyber defense solutions

In this section, we systematically investigate SDN-based attack prevention and mitigation methodologies, techniques, and procedures. We review the proposals in the literature and discuss them in the appropriate subsection according to the cyber threat categories given in Table 3. When selecting studies from the literature, we have considered the criteria listed below:

- Study must propose a methodology, technique, or procedure to defend networks using SDN. Papers that only review other studies have been excluded.
- Study must propose a prevention and/or mitigation strategy for at least one cyber threat category listed in Table 3. Our goal is to review and discuss how SDN capabilities can be used to protect networks proactively and respond to threats in an agile manner. To this end, studies proposing solely detection approaches for cyber attacks are largely excluded from our research. Studies that present approaches for detection along with prevention and mitigation are included and reviewed.
- Studies which primarily focus on new ways of attacking the SDN infrastructure have been excluded.
- SDN-based network defense approaches have not been commonly applied to cyber threats in the *Other* threat category in Table 3. Therefore, this category is out of scope in this work.

Each of the following subsections focuses on one type of threat and classifies relevant studies by defense type, applied SDN actions, and deployment locations/layers. If a study focuses on more than one cyber threat category, it is shown only in the most relevant subsection and a reference is given to the study in other subsections. Each subsection has a summary table with five columns: *name of study*, *defense type*, *description*, *approach*, and *deployment*.

- *Name of study*: The name of the proposed methodology, technique, or procedure is given in this column. If the study does not mention a name for the proposed approach, the reference number is used.
- *Defense type*: The proposed methodology, technique, or procedure is categorized as *prevention*, *detection*, or *mitigation*. More than one defense type can be used in a study.
- *Description*: A summary of the proposed defense mechanism is provided in this column.
- *Approach*: SDN-based defense approaches are classified in Table 4 under three main categories. One or more of the defense approaches listed in Table 4 are assigned to studies in summary tables in the following subsections.
- *Deployment*: Deployment locations of the proposed solution are listed in each table based on Fig. 3. Most solutions propose new modules/components in the architecture or modifications to existing components.

Table 2
Studies for cyber attack categorization and defined cyber attacks.

Focus of study	Defined cyber attacks
Attack vectors, attack targets, vulnerabilities [43]	Attack vector dimension (virus, worm, trojan, buffer overflow, DoS attack, spoofing, session hijacking, wireless network attack, web application attack, physical attack, password attack, information gathering attack, blended attack)
Attack vectors, targets, impacts [45]	Attack vector (misconfiguration, kernel flaws, design flaws, buffer overflows, insufficient authentication validation, insufficient input validation, symbolic link, file descriptor attack, race condition, incorrect permissions, social engineering)
End point threat categories [49]	Phishing, ransomware, DDoS, APT, privilege escalation, trojan, web application attack, blended threat, spyware, rootkits, man-in-the-middle attack, chained exploits, key logger, malware, kernel mode exploits
End point threat categories [50]	Targeted attacks (espionage, subversion, sabotage), email attacks (phishing, malware, spam), web attacks, ransomware
Taxonomy for network and computer attacks based on responses [51]	Infection phase (virus, worm, trojan), exploding (buffer overflow), probe (sniffing, port mapping, security scanning), cheat (IP spoofing, MAC spoofing, DNS (Domain Name System) spoofing, session hijacking, cross-site scripting (XSS), hidden area operations, input parameter cheating), traverse (brute force, dictionary attack, doorknob attack), concurrency (flooding, DDoS)
Methods of operations [52]	Operational impact (misuse of resources, user compromise, web compromise, malware, DoS)
Pattern based security threats for distributed systems [53]	Network communications attacks (i.e., passive and active eavesdropping, source spoofing, protocol sniffing, covert network channel, session hijacking), passing illegal data (injection), remote information inference (scanning, probing, information disclosure, data inference), uncontrolled operations (i.e., unauthorized access, spoofing privileged processes, process overflow)
Taxonomy of network security tools [54]	Trojans, DoS, DDoS, packet forging attack, browser attacks (XSS, XSS request forgery), server attacks (protocol attack, SQL injection, code injection, buffer error, URL misinterpretation), fingerprinting attack, user attack (user to remote, remote to local), sniffing, network scanning
Taxonomy for routing system intrusion detection [55]	Attack layer (sniffing, traffic flood, spoofing, MAC address table overflow, routing spoofing, DoS, management protocol attack, imprecise management), Attack vector (virus, worm, buffer overflow, physical attack, password attack, information gathering)

Table 3
Cyber threat categorization used in this study.

Category	Cyber threats
Scanning attacks	Network scanning, probing/fingerprinting attack [1,43,51,53,54]
Spoofing attacks	IP address spoofing, ARP (Address Resolution Protocol) spoofing, DNS spoofing, MAC (Media Access Control) address table overflow, routing spoofing, network management protocol attack, wireless network attack, SSL/TLS attack, man-in-the-middle attack [43,49,51,53–56]
Network-Level DoS attacks	DDoS, UDP flood, SYN flood, ping of death, teardrop attack, low-rate DoS attack, ICMP flood, DNS attack, traffic flood [1,43,45,49,51–55]
Sniffing attacks	Identity information sniffing, information gathering attack, information disclosure attack, covert network channel, credential compromise, eavesdropping, espionage [43,50,51,53–55]
Malware and social engineering attacks	Ransomware, worm, trojan, adware, key logger, spyware, virus, malicious scripts, browser attacks, spam, phishing, spear-phishing, whaling attack, URL misinterpretation attack [1,43,45,49–52,54–56]
Web application attacks	Application-level DoS attack, XSS, illegal input parameter attack, injection attack (SQL, command, LDAP injection), cross site request forgery (CSRF), authentication and session management attacks, session hijacking, misconfiguration exploits, brute force attacks, misuse of application [43,45,49,50,52–54,56]
Other (Hardware, Operating system, and Process attacks)	Memory-based attacks, privilege escalation, buffer overflow, kernel-mode exploits, rootkits, process exploits, hardware backdoor, APT, blended threats, chained exploits, password attack [1,43,45,49–52,54–56]

The reviewed studies follow different cyber defense strategies, which can be defined as the high-level planning and directing of cyber defense operations. In the SDN-based network security domain, defense strategies can be grouped under the following categories:

- *Policy Based Defense Strategy*: In SDN-based networks, it is easier to define security policies dynamically using system properties and network statistics than in traditional networks where most security mechanisms are managed using static security policies.
- *Machine Learning Based Defense Strategy*: Machine learning methods are used to detect attacks and generate security rules to respond to threats.
- *Moving Target Defense Strategy*: One or more network system properties are changed to make the attack surface unpredictable to adversaries.
- *Collaborative/Distributed Defense Strategy*: Multiple network domains collaboratively share cyber threats and defend against attacks.

The rest of this section presents a detailed review of SDN-based defense solutions against each threat category, along with an overall assessment and a summary table at the end of each subsection.

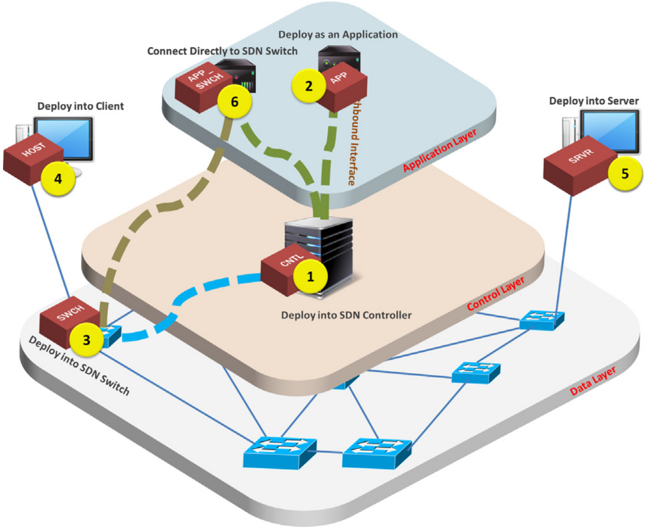


Fig. 3. Deployment locations of SDN-based defense solutions.

Table 4
Cyber defense approaches.

Category	Approach	Short name	Description
Traffic blocking	Drop packet	DROP	Drop packet according to match rule
	Blacklist	BLACKLIST	Drop packets which have source or destination address in blacklist
	Network separation	NETWORK-SEP	Drop all packets from/to specific host (quarantine) or manage separated networks
Network and data protection	Network access control	NAC	Unauthorized hosts or flows are not allowed, hosts must authenticate initially
	Rate limit	RATE-LIMIT	Change bandwidth of specific network traffic
	Change flow route	REROUTE-FLOW	Change routing path of specific network traffic
	Reroute on multiple paths	REROUTE-MPATH	Send parts of flows on different routes
	Deploy random decoys	RANDOM-DECOY	Deploy random honeypots or honeynets
	Encrypt payload	PAYLOAD-ENC	Encrypt payload of traffic
Deception	Different virtual view	VIRTUAL-VIEW	Generate different network views per host
	Modify header	HEADER-MOD	Update information in packet headers
	Modify address/port	ADDRESS-MOD	Change and maintain MAC addresses, IP addresses or port information in packets
	Modify payload	PAYLOAD-MOD	Change or distort information in payload data
	Send to decoy	SEND-TO-DECOY	Reroute traffic to honeypot or honeynet

4.1. Defense against network scanning attacks

Cyber attacks often start with reconnaissance to gather information about potential targets and to identify vulnerabilities. Scanning a network to identify active devices and open ports is a critical step in the reconnaissance effort. Maintaining a unique network view for each host (VIRTUAL-VIEW) through changing IP addresses or port information in packets (ADDRESS-MOD) and distorting information in payload data (PAYLOAD-MOD) are the most common defense approaches to deal with network scanning attacks. Among these, VIRTUAL-VIEW is the most comprehensive approach to tackle scanning. Achleitner et al. [16,57] propose a network deception system, named RDS (Reconnaissance Deception System), to defend against reconnaissance attacks. RDS blocks network reconnaissance by delaying the scanning actions of adversaries and invalidating their collected information. RDS manages different virtual network views for each host in the network, and takes actions for DNS, ARP, DHCP (Dynamic Host Configuration Protocol) and data transmission packets. The authors claim that RDS protects the network from reconnaissance attacks and can help to detect/respond to advanced persistent threats (APT).

Chiang et al. [58] present another adaptive cyber deception system called ACyDS. ACyDS provides different virtual network views to each host in the network dynamically in coordination with DNS and DHCP servers. The TTL field in IP headers is used to manage virtual views, and honeypots are deployed randomly into each virtual view. The authors argue that the proposed approach deters reconnaissance, prevents collisions, and increases the likelihood and confidence of detecting attackers. Robertson et al. [59] propose a similar deception system, named CINDAM (Customized Information Networks for Deception and Attack Mitigation) that prevents attacks by creating per-host views. They claim that the proposed system can make it harder to attack a network without affecting network operations and modifying client or server software. On the other hand, these two papers do not give detailed information about network reconnaissance or evaluate different attack strategies.

ADDRESS-MOD is another common defense approach against network scanning attacks. Jafarian et al. [60] present OpenFlow Random Host Mutation (OF-RHM) based on moving target defense to transparently mutate IP addresses with high unpredictability and rate. Their SDN controller frequently assigns a random virtual IP to each host. The real IP is not changed and the mutation is transparent to end hosts. Named hosts are reachable via the virtual IP addresses acquired from the DNS server. Only authorized entities can use real IP addresses of hosts. The authors report that OF-RHM can invalidate the information gathering of external scanners up to 99%, save up to 90% of network hosts from even zero-day unknown worms and other scanning-based attacks. Moreover, they claim that the proposed architecture defends against stealthy scanning and worms. Jafarian et al. [61] propose other techniques that enable host-to-IP binding of each destination host to vary randomly across the network based on the source identity and time. Source identity and time randomization will invalidate attackers' view of the network by causing the collected reconnaissance information to expire. In other works, Jafarian et al. [62,63] present IP address randomization techniques that adaptively reconfigure the addresses of network hosts in order to defeat scanning attacks transparently. With the proposed technique, fast and accurate hypothesis testing is conducted for characterizing adversarial behavior, and a very fast IP randomization is applied onto both legacy networks and software-defined networks.

Ma et al. [64] propose a self-adaptive endpoint hopping technique (SEHT) using moving target defense (MTD). The proposed technique is composed of a randomization controller, hopping switches, and hopping endpoint components managing Ethernet frames using TAP under Linux. Randomization controller monitors failed network requests, detects the type of network scanning strategy, selects the hopping strategy, and collaborates with hopping switch and hopping endpoint components to implement the selected hopping strategy for a predefined period. They claim that SEHT can thwart different types of scanning attacks. With a different perspective, MacFarland and Shue [65] present a host based moving target defense technique that tries to guarantee only for unmodified clients while avoiding scalability limitations. They evaluated their technique using unpredictability, vastness, periodicity, uniqueness, revocability, availability and distinguishability parameters. They argue that the suggested technique provides key security properties and allow defenders to distinguish between trustworthy and untrustworthy clients.

Wang et al. [66] propose a network defense method based on random domain name and address mutation (RDAM). Domain names of all hosts are periodically changed using dynamic domain name generation, which increases the scanning space of the attacker and reduces the probability of reaching targets using DNS query list and time window methods. They claim that the proposed method can thwart scanning attacks and worm propagation. In another study based on moving target defense, Wang and Wu [67] present Sniffer Reflector to defend against network reconnaissance. Their method reflects scan traffic to a shadow network where scan replies are generated and obfuscated using virtualization. They argue that Sniffer Reflector is effective and efficient in spoiling various network reconnaissance attacks.

Zhao et al. [68] claim that fingerprinting attacks are one of the most severe network threats. These attacks aim to obtain information about target hosts to make preparations for future attacks. The authors propose SDN-based fingerprint hopping method (FPH) to defend against fingerprinting attacks. FPH uses the moving target defense approach to show a hopping fingerprint toward attackers. A game model is defined in the study and used to model fingerprinting attacks. An optimal defense strategy is generated with the equilibrium of the game. Kampanakis et al. [69] propose methods for host/port scanning attacks, and operating system or service fingerprinting attacks. Each packet is buffered to ensure consistent behavior, returned random answers for requests, and added random delays in proxied TCP handshakes. The authors claim that the proposed methods result in a large delay overhead for attackers attempting service discovery and 100%–200% overhead for port scanning and operating system fingerprinting attacks.

Shin et al. [70] propose a framework called FRESCO to develop security applications using SDN and present security applications for scanning attack, malware, botnet detection and mitigation. Using the FRESCO scripting language, applications can define constraint-based actions for flows such as drop, forward, redirect, mirror, network separation, etc. The authors report that FRESCO introduces minimal overhead and enables rapid development of popular security functions with fewer lines of code.

Scanning is usually a precursor to other types of attacks. Cabaj et al. [71] present an SDN-based security framework to detect and mitigate TCP SYN-based scanning attacks on the Internet of Radio Light (IoRL) system designed according to 5G architectural requirements. As such systems become more widespread over time, SDN will be a powerful tool for developing solutions that can meet the stringent security requirements of 5G.

Highlights: Scanning is a very important step in complex attacks because it provides the attacker with valuable information about the target network. A common goal in the defense against scanning is to either obfuscate the attacker's view of the network or deceive them into attacking the wrong target. To this end, deception-oriented techniques such as moving target defense and honeypots are popular as preventive measures in the reviewed studies. In terms of the defense strategy, 11 of the 16 proposals discussed in this subsection follow a moving target defense strategy. As seen in Table 5, most proposals use at least one of the following three defensive actions: presenting virtual views to users, deploying decoys, and packet modifications.

Defensive mechanisms against scanning are threatened by a new paradigm called adversarial network forensics [34], which infers the details of flow rules in an SDN with high accuracy using only a few probing packets. With this kind of information, attackers would be able to craft more dangerous attacks that can defeat the countermeasures. Since this approach to scanning is a fairly recent development, the literature lacks effective methods to neutralize it. New defense proposals should be aware of the fact that the SDN infrastructure may be susceptible to adversarial network forensics, and devise methods to make networks resistant to it.

4.2. Defense against spoofing attacks

Spoofing typically takes advantage of the lack of authentication in major networking protocols, and is commonly employed in attacks such as DDoS and man-in-the-middle. Sahri and Okamura [72–74] focus on DDoS attacks targeting DNS servers. These attacks usually perform IP address spoofing and are hard to detect due to the difficulty of distinguishing a legitimate query and an attacker for the DNS provider. The authors present an SDN based mechanism, named CAuth, that autonomously blocks the spoofed query packet while authenticating legitimate DNS queries. When a query is sent to a server, an authentication packet is sent back to the client by the controller, which then replies with an authentication packet back to the controller. The controller only forwards the query to the DNS server if it receives the authentication packet from the client. This mechanism is designed with no changes in the existing DNS application and OpenFlow protocol. The authors argue that their method can differentiate between legitimate and attack packets before they reach the DNS server using only 1.2 times higher bandwidth on average compared to the situation with no such protection.

Several works focus on ARP spoofing and propose solutions to defend against it. Masoud et al. [75] highlight that many network attacks such as man-in-the-middle, DoS, and session hijacking start with ARP spoofing and propose an algorithm to solve this problem using SDN. The algorithm considers dynamic and static IP address assignments and verifies IP addresses. Cox et al. [76] present an SDN security module, called Network Flow Guard for ARP (NFGA) that augments MAC-learning protocols on OpenFlow-enabled switches. The module hashes the physical address of a node with dynamic or static IP and port association to deny ARP spoofing in real time. They claim that no change is required in the network topology or protocols, and no client software installation is needed.

Nehra et al. [77] present a method for verification and detection of ARP based attacks, named Traffic Pattern Based Solution to ARP Related Threats (FICUR). In the proposed method, a customized SDN controller collects required network parameters and analyzes these parameters to verify and detect the attacks. Moreover, mitigation is also performed on the fly using SDN capabilities. The authors evaluate their method on both simulated and real environments to conclude that it adds a limited overhead to the network. Ubaid et al. [78] point out that a limited number of SDN devices can be deployed among legacy devices in organization networks because of budget constraints and limited skills. They propose a technique to automatically detect the attack condition and mitigate ARP spoofing in hybrid SDN. They adopt graph based traversal mechanisms to indicate the location of the attacker. Alharbi et al. also [79] present a mitigation approach against ARP spoofing attacks leveraging the centralized network control of SDN. ARP request and reply packets are processed to verify requests in this method.

Like ARP, NDP (Neighbor Discovery Protocol) in IPv6 also lacks authentication, and is vulnerable to spoofing attacks. Lu et al. [80] mention that NDP is easily spoofed without protection, and propose an SDN-based authentication mechanism to verify the identity of NDP packets transmitted in a LAN. This mechanism uses the centralized control and programmability capabilities of SDN, and needs no additional protocol supporting or configuration at hosts and routers. The authors argue that the proposed mechanism can effectively prevent spoofing attacks and other derived attacks based on spoofing.

Mattos and Duarte [81] propose an authentication and access control mechanism, named AuthFlow, based on host credentials. AuthFlow applies IEEE 802.1X in an OpenFlow network for low overhead and fine-grained access control based on host privileges. AuthFlow has a framework which enables SDN controllers to use the host identity as a new flow field to define flow table rules. The authors develop a prototype for the proposed mechanism using an SDN controller. They state that AuthFlow denies access of hosts either without valid credentials or with revoked authorization, and prevents unauthorized hosts from accessing network resources. Moreover, the proposed mechanism can be used to tackle scanning and sniffing attacks.

Table 5
SDN-based defense solutions against network scanning attacks.

Research	Defense type	Proposed methodology, Technique, or Procedure	Approaches	Deployment
RDS [16,57]	Prevention, Detection	Network view is changed for each host. Malicious flows which try to connect to honeypots or vulnerable hosts are detected as scanning attack.	VIRTUAL-VIEW, RANDOM-DECOY	CNTL, APP
ACyDS [58]	Prevention, Detection	The TTL field in IP headers is used to manage virtual views. Honeypots are deployed randomly into each virtual view. Malicious flows which try to connect to honeypots are detected as scanning attack.	VIRTUAL-VIEW, RANDOM-DECOY	CNTL, APP
CINDAM [59]	Prevention, Detection, Mitigation	Network view is changed for hosts. Hosts which send packets to IP addresses of honeypots are detected as attacker.	VIRTUAL-VIEW, RANDOM-DECOY	CNTL
OF-RHM [60]	Prevention	A random virtual IP is assigned for each host frequently. Virtual IP is translated to/from the real IP of the hosts are accessed using virtual IP addresses acquired via DNS.	ADDRESS-MOD	CNTL, APP
[61]	Prevention	Host-IP binding of each destination host is applied randomly across the network based on the source and time.	ADDRESS-MOD	CNTL, APP
[62]	Prevention	IP address is changed and new virtual IP address is requested from DNS server.	ADDRESS-MOD	CNTL
RHM [63]	Prevention	Real IP addresses are periodically transformed to ephemeral IP addresses in SDN switches. Direct access to hosts using real IP addresses is authorized based on access control policy of network. Reverse-DNS name mutation and MAC address mutation are generated.	ADDRESS-MOD, MAC-MOD	CNTL
SEHT [64]	Prevention, Detection, Mitigation	IP address and port of end points are changed using SMT randomly in a predefined period. Entropy of network traffic is calculated to discriminate scanning strategy.	ADDRESS-MOD	CNTL, SWCH, HOST, SRVR
[65]	Prevention	Clients must engage with DNS server to actually reach synthetic IP address of the legitimate server.	ADDRESS-MOD	CNTL, SWCH, HOST, SRVR
RDAM [66]	Prevention	Clients must authenticate to authentication server using proof-of-work (PoW) schemes. Authentication server manages different dynamic domain name lists to authorized clients according to their authority.	NAC	CNTL, SWCH
Sniffer Reflector [67]	Detection, Mitigation	Customized Snort application detects possible scan traffic. Scan traffic from source IP is redirected to virtual shadow network.	SEND-TO-DECOY	APP, SWCH
FPH [68]	Detection, Mitigation	IDS detects fingerprinting behavior in traffic. Fingerprinting attack and its defense is modeled as a signaling game (Perfect Bayesian Equilibrium). If defender's belief for traffic is suspected, packets in traffic is tagged. Fingerprints in tagged packets are modified and forwarded as response to request.	PAYLOAD-MOD	CNTL, APP
[69]	Prevention	Fake responses are sent to scanning attack for status of port or operating system fingerprinting.	PAYLOAD-MOD	CNTL
FRESCO [70]	Detection, Mitigation	Conditional filters can be applied to network. Programmable actions can be defined to flows like drop, forward, redirect, mirror, network separation, header modification.	REROUTE, DROP, NETWORK-SEP, HEADER-MOD	CNTL, APP
[71]	Detection, Prevention	State of each TCP connection in the traffic handled by the switch is tracked. Number of pending state connections are maintained.	DROP	CNTL

Kuliesius and Dangovas [82] mention that user box spoofing and security device bypassing problems can be seen in networks, even though network security actions are applied like authentication, authorization, and accounting. They propose a model to secure the access control system by complementing the network with elements based on SDN. Their model introduces controller modules dedicated to authentication, registration and tracking of network devices, hosts and users, management of data flows, and mobility of users/hosts. The access layer switches are authenticated, mapped to the network topology and tracked. It is possible to bind users to the appropriate switch/port unambiguously and effectively control their traffic policy. Moreover, the proposed model can thwart scanning attacks.

Kwon et al. [83] propose an anti-spoofing mechanism called BASE (BGP-based Anti-Spoofing Extension). BASE is an SDN-based anti-spoofing protocol designed to fulfill the incremental deployment properties, which are defined as initial benefits for early adopters, incremental benefits for subsequent adopters, and effectiveness under partial deployment. BASE is based on three techniques: message authentication code (MAC), one-way hash chains and packet marking. MAC and one-way hash chains are used to generate unique values for a filter node, and packet marking is used to store and send the value to destinations. The authors argue that BASE shows desirable IP spoofing prevention capabilities under partial deployment and just 30% BASE deployment can drop about 97% of spoofed packets.

Some research efforts rely on Source Address Validation Improvement (SAVI) to combat IP spoofing. Liu et al. [84] present an SDN controller module called SDN-SAVI to enable SAVI functionalities in SDN networks. They express three challenges in SDN-SAVI design. First, switch ports are classified into three categories and each port is assigned to these categories to avoid redundant address assignment mechanism snooping and address binding on all switches. Second, address assignment mechanism packets are limited and controller's verification job is delegated to switches to prevent resource exhaustion attacks. Third, multiple tables defined in OpenFlow are used to solve the issue of flow table explosion. Yao et al. [85] propose a protective perimeter approach named Virtual Source Address Validation Edge (VAVE) to improve the SAVI solutions. A packet originating from outside the perimeter will be redirected to the SDN controller when it reaches the perimeter. The controller checks the validity of the source IP based on generated rules. Packets that have forged source IPs are blocked for a period. In another work, Yao et al. [86]

propose a framework, called SEFA (Software dEfined Filtering Architecture), for route-based IP spoofing filtering. SEFA is based on SDN switch and controller modification to use OpenFlow+, a customized version of the OpenFlow protocol. The authors report that SEFA is able to reduce complexity, overhead, and latency of filtering rule generation and installation for spoofing.

Highlights: Spoofing is a type of attack that can be carried out at multiple network layers on different protocols such as ARP, IP, DNS, NDP, etc. Nearly all defense mechanisms against spoofing given in Table 6 rely on some sort of network access control (NAC) involving steps like authentication and source verification. Switches and the controller play a key role in many of these solutions. However, network functions like deep packet inspection (DPI) can provide assistance against spoofing, so employing DPI as part of a service function chain for suspicious traffic to detect spoofing is an idea worth exploring more.

4.3. Defense against network level DoS attacks

SDN provides an arena where the potential impact of DoS attacks and the efficacy of defense approaches are both elevated. There are many different approaches to combat DoS in SDN, ranging from proactive resource management and attack deception to anomaly detection based response mechanisms.

Some methods rely on network agility to prevent attacks or reduce their impact. Jafarian et al. [87] propose an agile multipath routing approach, called random route mutation (RRM) which combines game theory and constraint satisfaction optimization to determine the optimal strategy while satisfying security, performance and QoS requirements of the network. RRM models route selection as a constraint satisfaction optimization and formalize it using Satisfiability Modulo Theories (SMT) to identify efficient practical routes. The authors provide algorithms for sound and smooth deployment of RRM on conventional as well as software defined networks. They state that RRM can protect up to 90% of flow packets from being attacked against persistent attackers, as compared with single path routing schemes. Gillani et al. [88] propose a model named Agile Virtual Network Framework Model to defend against DDoS attacks by proactively changing the infrastructure of critical resources. Agile VN Framework uses virtual networks to reallocate network resources dynamically using VN placement and migrates infrastructure to new resources while maintaining network integrity.

Fichera et al. [89] propose a protocol and an application against SYN flood, named OPERETTA (OPEnflow based REmedy to TCP SYN flood Attacks). OPERETTA is implemented in the SDN Controller, and manages incoming TCP SYN packets and rejects fake connection requests. The evaluation concludes that CPU and memory consumption of OPERETTA is low, and OPERETTA is resilient to TCP SYN flood attacks. In another attempt to prevent SYN flood attacks, Shin et al. [90] introduce AVANT-GUARD, composed of connection mitigation and actuating triggers components. Actuating triggers are located in between SDN Controller and SDN switches to increase both detection of, and response to, the changing traffic rate. These triggers are used to register for asynchronous call backs to upper layer applications and to insert conditional flow rules only activated when a trigger condition is detected.

Giotis et al. [91] propose a detection and mitigation architecture that uses OpenFlow and sFlow [114] capabilities. Anomalies are detected using entropy values calculated from network traffic, collected in 30 s time intervals using sFlow. Their mitigation approach involves managing a whitelist and blocking other network traffic. OpenFlow is used to mitigate attacks via flow table modifications. The authors report their experiments for DDoS, worm and port scanning attacks. Hussein et al. [92] propose an SDN security design approach to prevent DDoS attacks and trace back the source of the attack. The proposed approach introduces a new SDN plane, the security plane, in addition to the data plane and parallel to the control plane of SDN. The security plane has a third-party agent on the switch and a third-party software module alongside the controller. The authors report that the proposed system enforces different levels of real-time user defined security with low overhead and minimal configuration.

Joldzic et al. [93] present a distributed, scalable solution called Transparent Intrusion Detection System (TIDS) for detecting and preventing lower-level DoS attacks. With features of SDN, scalable active traffic balancing among multiple traffic processors is applied based on device polling. TIDS tracks a larger number of attacked hosts, and mitigates all active attacks simultaneously. After the source and destination of the attack have been isolated, the processor sends a Flow Modification Request message containing the list of addresses to be banned for a limited amount of time. The authors claim that the proposed system is transparent to the external devices, and invalid TCP sequence numbers, SYN floods and other similar attacks can be detected easily without additional resource consumption. Li et al. [94] propose a traffic engineering system called DrawBridge to enable communication between hosts and internet service providers (ISP). DrawBridge is an SDN controller that can push flow rules to SDN switches in an ISP, or communicate with another DrawBridge controller in the ISP upstream. The authors claim that the proposed system allows a controller to verify and process the flow rules, and deploy them at SDN switches or upstream ISPs with the best location for filtering DDoS traffic.

Miao et al. [95] present SDN-based NIMBUS framework, comprising a set of VM instances that analyze traffic for attack detection and an auto-scale controller. The presented framework scales VM instances to avoid overloading, and dynamically instantiates anomaly detector and mitigation modules. The authors highlight that blacklist or rate limiting strategies are applied on the detected attack traffic at the routers. Oktian et al. [96] introduced an application named Dossy on the application layer of SDN to mitigate DoS attacks along with IP/MAC Spoofing and bulky/garbage message attacks in OpenFlow local networks. The application keeps track of nodes in the network. Initially, the SDN controller collects MAC and IP information of all hosts and binds them together in a hash table. Every packet-in message is processed, and IP/MAC addresses are analyzed considering managed hash table. Dossy reduces the throughput of the controller by 7.76% and increases the latency by 7.37% compared to default learning switch.

Scalability, fast reaction and low overhead are necessary features for any DDoS defense mechanism. Piedrahita et al. [97] propose FlowFence, a lightweight and fast DoS detection and mitigation system. The SDN controller coordinates bandwidth assignment of controlled links and limits the flow transmission rate along a path to prevent starvation of nodes. Flows exceeding the assigned limits are penalized. They argue that FlowFence avoids users' starvation of network resources while adding a small overhead. Another proposal that uses rate limiting as the main response is SDNManager [98], where flow information is continuously monitored on the network and bandwidth demands are estimated for the future. While estimating the bandwidth, the autoregressive conditional heteroscedasticity (ARCH) model generally used in econometrics is adopted. Using this model, changes in time series are measured and the volatility of the total bandwidth of the network is estimated. After estimation, the nodes that are above the estimated bandwidth of the current period are penalized to the extent that they exceed the bandwidth.

Wang et al. [99] propose a cloud based DDoS attack mitigation architecture, named DaMask, to enable attack detection and

Table 6
SDN-based defense solutions against spoofing attacks.

Research	Defense type	Proposed methodology, Technique, or Procedure	Approaches	Deployment
CAuth [72–74]	Prevention	DNS requests are buffered for a few seconds and authentication packet is sent for each DNS request. If client responds to authentication packet, DNS request is allowed.	NAC, DROP	SWCH
[75]	Detection, Mitigation	Dynamic or static IP:MAC address pairs are verified. Unverified ARP packets are dropped.	NAC, DROP	APP
NFGA [76]	Detection, Mitigation	MAC address with IP:port is maintained at real time using Pyretic. Invalid ARP packets are dropped.	DROP	APP
FICUR [77]	Prevention, Detection, Mitigation	All ARP traffic is directed to the controller and verified using log file. IP packets, ARP request and reply packets are processed to verify. Packets for irrelevant ARP data are blocked.	NAC, DROP	SWCH
[78]	Prevention, Detection, Mitigation	All ARP traffic is directed to the controller and verified. IP and MAC addresses of ARP packets are verified using topology information. Unverified ARP packets are dropped.	NAC, DROP	SWCH
SARP-NAT [79]	Detection, Mitigation	ARP request and reply packets are processed to verify. ARP request packets are queued in pending queue and only ARP response packets in pending queue are allowed.	NAC, DROP	SWCH
[80]	Prevention, Detection, Mitigation	All NDP traffic is directed to the controller and verified using network topology information. Unverified NDP packets are dropped.	NAC, DROP	SWCH
[81]	Prevention	EAP authentication according to IEEE 802.1X is performed in the network.	NAC	SWCH, APP
[82]	Prevention	User and access rights of flows are checked and flows are created only for legitimate traffic using authorization databases (i.e., devices and users, group rights and flow counters).	NAC, DROP	SWCH
BASE [83]	Prevention	ToS (Type of Service) field of IP header is used as flow marking field, and used to route flows in autonomous systems.	REROUTE-FLOW, HEADER-MOD, DROP	SWCH
SDN-SAVI [84]	Prevention	Each IP address has a state. If state of IP address changes to BIND, a binding entry is created using IP address, switch and switch port. If the state changes to NO-BIND, the IP address is removed from the binding table.	NAC, DROP	SWCH
VAVE [85]	Detection, Mitigation	Controller checks whether packet source is valid based on generated rules. If source is detected as spoofing IP, packet is dropped.	NAC, DROP	SWCH
SEFA [86]	Prevention	Route-based IP spoofing mechanism checks each source IP with the expected incoming interface. Route-based forwarding is applied with coordination of controller and switches.	NAC, DROP	SWCH, APP-SWCH

to allow fast and specific attack reaction. DaMask is composed of DaMask-D (network attack detection system) and DaMask-M (attack reaction module). If an alert is received by DaMask-M, module searches a countermeasure for the alert. The default countermeasure is to drop the packet if there is no preset match for the alert. DaMask-M has as a set of common APIs so that different defense countermeasures can be customized for different DDoS attacks. Three basic countermeasures are defined as follows: forward, drop and modify. If the countermeasure is selected, DaMask-M pushes the flow rules to the switch through the SDN controller.

FloodDefender [100] was proposed as a framework for protecting the SDN controller from DDoS attacks. It sits between the controller and the applications to detect attacks bombarding the controller with packet-in messages (generated when there is a flow table miss at a switch), filter attack packets, and manage flow rules. FloodDefender also protects a victim switch from the exhaustion of its control channel bandwidth by sending some of the packet-in messages to its neighbor switches, which will then forward them to the controller. It is reported that both the CPU utilization at the controller and flow table utilization at the switch stays below 15% during an attack, meaning that FloodDefender is effective.

Several different statistics regarding packet counts and byte counts are held by SDN switches. These statistics can be continually collected by the controller and processed to detect DDoS attacks. For instance, some studies perform DDoS detection by analyzing the distribution of packet-in counts [101] or packet-in/transmitted packet ratios [102]. Such methods are particularly effective against DDoS attacks indirectly targeting the controller via the southbound interface. Some other systems make use of the statistic collection capability in SDN to get the desired information from switches. SoftGuard [103], a solution for low-rate DoS attacks, works by adding monitoring rules to each switch to periodically measure the total amount of data matching these rules and continuously monitor the total bandwidth. When a significant decrease in bandwidth occurs, the attacker identification stage begins. During the detection phase, switch interfaces above the average bandwidth are marked if they are similar to the predetermined bandwidth decrease. As a method of combating the attackers, the authors proposed reducing their bandwidth or adding drop rules.

Some studies propose defense systems to detect and mitigate link flooding attacks (LFA) such as the crossfire attack [115]. LFA applies legitimate-looking low density flows to flood a group of selected links, and is hard to distinguish by traditional schemes. Wang et al. [104] propose a scheme called Woodpecker to mitigate the LFA. Woodpecker contains a heuristic algorithm for incremental SDN deployment, a mechanism for LFA detection and a centralized traffic engineering module. The proposed scheme finds the average and minimum number of paths between the source–destination node pairs to evaluate the connectivity of a network. These additional paths will be used to balance the traffic. They evaluate the scheme and report that bandwidth utilization of LFA attacked links is reduced by around 50% and average packet loss rate and jitter are effectively mitigated. Another proposal called LFADefender [105] leverages SDN to achieve cost-efficiency and flexibility. LFADefender examines the network to identify potential target links with high flow density, continuously monitors link congestion, reroutes traffic away from

Table 7
SDN-based defense solutions against network-level DoS attacks.

Research	Defense type	Proposed methodology, Technique, or Procedure	Approaches	Deployment
RRM [87]	Detection, Mitigation	Random route is defined using game theory and constraint satisfaction optimization to determine the optimal strategy for attack. Flows are rerouted using selected routing paths.	REROUTE-FLOW	CNTL
[88]	Mitigation	The migration mechanism executes VN migration strategy generated by the framework on a virtualized infrastructure.	REROUTE-FLOW	APP
OPERETTA [89]	Prevention	SYN packets are sent to controller, which responds with a SYN-ACK to client host without sending to target host.	NAC	CNTL
AVANT-GUARD [90]	Prevention, Detection, Mitigation	Packet and flow statistics are used to determine if loaded conditions are satisfied. Conditional flow rules are inserted when related condition event is triggered.	HEADER-MOD, DROP, BLACKLIST, REROUTE-FLOW, NAC	SWCH
[91]	Detection, Mitigation	Flow info collected from sFlow is used to detect anomaly using entropy-based algorithm. Flow entries with higher priority are inserted in order to block malicious traffic.	DROP	APP-SWCH
[92]	Prevention, Detection, Mitigation	Switches send packets to detection engine using Pyretic. Detection engine sends the appropriate rules for the controller to insert into the switches to block the attack.	DROP	SWCH, APP-SWCH
TIDS [93]	Detection, Mitigation	Device polling processor analyzes flows to detect anomalies and sends a Flow Modification Request containing the list of addresses to be banned for a predefined time.	DROP	APP-SWCH
DrawBridge [94]	Mitigation	Flow rules are pushed to SDN switches in ISP network, or sent to SDN controller in ISP to apply rules.	REROUTE-FLOW, DROP	CNTL
NIMBUS [95]	Detection, Mitigation	Traffic monitoring in cloud data center typically employs uniform sampling with very low sample rate. Blacklist or rate limit with auto scalable VMs	RATE-LIMIT, BLACKLIST	CNTL
Dossy [96]	Detection, Mitigation	Both flow based and packet-in analysis are used to detect DoS attack. Flow mode messages are sent into switches to block DoS attack addresses	DROP	CNTL
FlowFence [97]	Detection, Mitigation	Traffic congestion is detected by SDN switches. SDN controller control bandwidth for flows	RATE-LIMIT	CNTL, SWCH
SDNManager [98]	Prevention	Monitors traffic, estimates bandwidth demands and punishes flows exceeding estimates.	RATE-LIMIT	CNTL
DaMask [99]	Detection, Mitigation	Anomaly based attack detection using graphical model. Countermeasure uses forward, drop and modify actions.	HEADER-MOD, DROP	APP
FloodDefender [100]	Detection, Mitigation	Detects attack using packet-in messages, filters attack packets, and manages flow rules.	HEADER-MOD, DROP	APP, CNTL
SECOD [101]	Detection, Mitigation	Drops packets based on packet-in thresholds.	DROP	CNTL
[102]	Detection, Mitigation	Monitors the fairness of packet-in/tx distribution of hosts, and detects attack when the fairness drops significantly.	BLACKLIST	CNTL
SoftGuard [103]	Detection, Mitigation	Installs flow rules on switches to detect low-rate TCP attacks, and performs mitigation on ingress switches.	RATE-LIMIT, DROP	CNTL, SWCH
Woodpecker [104]	Detection, Mitigation	Switches notify about congestion. Controller reroutes attack traffic to decoy servers, balances congested link flows with alternate paths, or drops packets using a blacklist.	SEND-TO-DECOY, REROUTE-FLOW, DROP	CNTL, SWCH
LFADefender [105]	Detection, Mitigation	Examines the network to identify potential target links, monitors link congestion, reroutes traffic away from congested links, and blocks malicious traffic.	REROUTE-FLOW, DROP	CNTL
RADAR [106]	Detection, Mitigation	Suspicious traffic is throttled and attack traffic is dropped using port-based max–min fairness technique.	RATE-LIMIT, DROP	CNTL
[107]	Detection, Mitigation	Classifies traffic using deep learning to detect DDoS, and installs drop rules on switches	DROP	CNTL
TDDAD [108]	Detection, Mitigation	Analyzes the temporal distribution of flow table hits and detects attacker using a neural network.	DROP	CNTL
Cochain-SC [109]	Detection, Mitigation	Entropy-based flow anomaly detection and mitigation using sFlow in intra-domain, smart contracts in Ethereum blockchain in inter-domain.	RATE-LIMIT, BLACKLIST, DROP	APP-SWCH, APP
[110]	Mitigation	IP blacklists are maintained using smart contracts in Ethereum blockchain collaboratively.	BLACKLIST, DROP	APP
Co-IoT [111]	Mitigation	Illegitimate IP addresses are managed using smart contracts in multiple domains.	BLACKLIST, DROP	APP
FastFlex [112]	Detection, Mitigation	In the data plane, the detector propagates the alarm across the network using probe packets. Rerouting, dropping, and IP obfuscation mitigation approaches are applied in switches for different modes.	REROUTE-FLOW, HEADER-MOD, DROP	SWCH
Poseidon [113]	Detection, Mitigation	Policy enforcement engine installs high-level defense primitives into programmable switches. Defense primitives are applied in the data plane.	DROP, BLACKLIST, RATE-LIMIT	SWCH

congested links to mitigate the effect of an ongoing attack, and blocks malicious traffic. Each of the above tasks is handled by a separate module on top of the Floodlight controller [116]. Moreover, Zheng et al. [106] propose an architecture called RADAR to detect and throttle DDoS attacks using adaptive correlation analysis. RADAR has detection modules for link flooding (including crossfire), SYN flooding, and DNS amplification attacks. RADAR analyzes traffic captured at various locations, throttles traffic if it is classified as suspicious, and drops packets using a port-based max–min fairness technique. The authors have evaluated their architecture on a real testbed and claim that RADAR can effectively and efficiently detect various attacks within short delays.

Deep learning has been proposed as a detector for DDoS in SDN environments [107]. As the controller has a clear view of the network and sees every packet that does not belong to a previously known flow, it is able to process packets to extract features from header fields and run a deep learning model to classify traffic as legitimate or DDoS. The controller can then quickly react to install flow rules to drop attack packets on switches. In another recent study, the temporal distribution of flow table hits was analyzed and attacker behavior was characterized using a neural network [108]. The approach is proved to be feasible, however, the authors state that the availability of legitimate services may be adversely affected.

In recent years, blockchain has been utilized in the defense against DDoS attacks. Singh et al. [117] review and evaluate blockchain-based DDoS mitigation architectures in the literature. SDN can be used to mitigate DDoS attacks within a single domain, however, blockchain and SDN together can be more effective in defending against DDoS attacks in the inter-domain setting. El Houda et al. [109] propose CoChain-Sc, a multiple SDN domain solution consisting of an intra-domain machine learning based DDoS detection and mitigation module, and an Ethereum blockchain based module used for inter-domain DDoS mitigation. In addition, two other studies [110,111] propose DDoS mitigation modules which send illegitimate IPs and get a list of other illegitimate IPs from an Ethereum blockchain using different smart contracts.

The recent literature also contains anti-DDoS proposals residing in the data plane to achieve better network performance compared to controller-based or middlebox-based solutions. For instance, FastFlex [112] relies on the concept of *multimodal data plane* to mitigate attacks at programmable switches, thereby eliminating the bandwidth overhead and delay associated with controller communication during an attack. Similarly, Poseidon [113] reduces overhead through the use of modularized defense primitives to deal with dynamic high-volume DDoS attacks. The Poseidon language, runtime, and orchestration component together offer an adaptable solution that can express a wide variety of DDoS defense policies and respond to attacks in an agile manner.

Highlights: The relationship between SDN and DoS is a fascinating topic because while the SDN architecture causes new risks due to centralization of control and programmability, it also presents good opportunities for improved DoS defense. Hence, the development of SDN-based anti-DoS techniques and systems has been a popular research topic in recent years. Table 7 provides a summary of the related studies reviewed in this paper. We see that the literature has grown more mature in this area because there are a wide variety of proposals: some studies focus on a particular DoS attack type while others present more general approaches, a total of seven basic defensive actions are employed, the distribution of defense types is close to uniform, and deployment locations include all three layers of SDN.

Some well-known DoS attack types such as SYN flood and link flooding have multiple solutions developed specifically against them. The most common defensive primitive is dropping packets, followed by flow rerouting and rate limiting. Many proposals take advantage of the centrality of the controller to collect information from switches and use different techniques to detect DoS attacks, among which threshold-based methods are popular. In prevention and mitigation, solutions are more effective when they use multiple defensive actions together, like rerouting traffic and rate-limiting certain protocols to reduce congestion in addition to blocking hosts according to a blacklist. Also, trying to anticipate the actions of the attacker is a good idea, as seen in LFADefender [105] which predicts target links from the network topology.

Despite the abundance and variety of work in this area, there are still some common issues in many of the studies. First, most solutions are realized at the controller by adding more modules to the software and more steps to the network control workflow. Such proposals will likely be unable to address the fundamental weakness of the SDN architecture against DoS attacks and may even exacerbate the problem as they place more burden on the controller. Some recent works in the literature [112,113] take steps in the direction of removing the dependence on the controller to achieve reduced overhead and improved performance by directly involving the data plane in network defense, which is worthwhile to be explored further in future studies. Second, it is difficult to find many studies with an evaluation that is both comprehensive and realistic. Some works present many results on a wide range of performance metrics using a very small test network lacking any realism, while others try to build a larger topology on simulation tools but run into some issues due to the limitations of the environment. In fact, the second problem hinders progress in solving the first one: without a strong evaluation, one cannot fully assess the benefit and the overhead associated with a new system, so the practical applicability of a given solution usually remains in question.

4.4. Defense against sniffer attacks

Sniffer attack is one of the most important attacks on network communication security, and the static nature of networks makes this attack easier. Attackers are able to eavesdrop the communication link in the network, monitor network status, and steal sensitive data. Zhao et al. [118] propose an SDN-based double hopping communication (DHC) approach to tackle sniffer attacks. The proposed approach dynamically changes endpoints in network packets and the routing paths. The traffic is distributed to multiple flows and transmitted along different paths. In addition, data from multiple nodes are combined so that attackers cannot obtain and recover the communication data. The authors conclude that DHC increases the overhead of sniffer attacks and difficulty of communication data recovery.

Duan et al. [119] present a proactive technique named Random Route Mutation (RRM) calculated using SMT on a defined overlay network. RRM enables randomly changing the routes of multiple flows in a network simultaneously to defend against reconnaissance, eavesdropping and DoS attacks. Liu et al. [120] also use RRM to develop an SDN-based architecture to counter network reconnaissance. Volume measurements and characteristic measurements of network traffic are used to detect attacks using a constructed entropy matrix. RRM is triggered according to the result of network anomaly detection or after a predefined period. The generation of a random routing path is performed using an improved ant colony algorithm as a 0–1 knapsack problem. They claim that the proposed method increases the difficulty of reconnaissance and eavesdropping, and reduces the impact of DoS attacks.

Furukawa et al. [121] propose a network mechanism that can prevent man-in-the-middle attacks and network eavesdropping by applying a new network address translation method. The proposed mechanism relies on multiple actions performed inside an SDN switch to strengthen the security level according to user requests. Ma et al. [122] address the problem of private data protection in network communication against eavesdropping and propose a moving target defense (MTD) method to thwart eavesdropping attacks by utilizing the protocol customization ability of protocol-oblivious forwarding (POF). Customization includes full protocol stack randomization, message packaging randomization, and routing path randomization. Nodes use dynamic message packaging and dynamic routing paths to protect their communication secret. The authors claim that the proposed strategies increase the difficulty of reorganizing message content for an eavesdropper and reduce the probability of intercepting session message packets.

There is a significant interest in SDN-based architectures for smart grids to improve network flexibility and security [36]. Germano Da Silva et al. [123] present an SDN application layer module that aims to prevent a possible eavesdropper from fully capturing communication flows between SCADA components. The mechanism uses dynamic and static multipath routing to modify communication routes between SCADA devices frequently, increases the privacy of the information in SCADA networks, and makes it more difficult for attackers to capture flows between SCADA devices. The authors perform an experimental evaluation and report that dynamic rules with a shorter lifetime make it more difficult for an attacker to intercept flows, but increase the management overhead in the controller.

Strong and reliable authentication is important to keep potential adversaries out and prevent sniffing attacks on networks. Villain et al. [124] propose an authentication mechanism using an SDN controller. Requests from an unauthenticated user who tries to access the web are redirected to a portal using HTTP redirection. After the portal authenticates the user, it notifies the SDN controller to install flow rules for the user.

Highlights: The studies reviewed in this subsection commonly aim for sniffer attack prevention using techniques like multipath routing and payload encryption/modification, as shown in Table 8. The moving target defense strategy is fairly popular against sniffing: 3 of the 7 studies analyzed in this subsection follow an MTD-based approach. The flexibility and ease of customizing the forwarding behavior in SDN is particularly useful for implementing multipath routing solutions: the controller takes care of all the required rule changes and forwarding continues smoothly without interruption. Yet, it may still be difficult to apply multipath routing in hybrid networks where SDN is only partially deployed. Therefore, the literature needs more studies analyzing the practicality of security solutions which depend on multipath routing in hybrid networks, and researchers should consider the case of partial SDN deployment and design methods to still be effective in such a situation.

4.5. Defense against malware, social engineering and web application attacks

Cabaj and Mazurczyk [125] propose two real time mitigation methods for ransomware based on behavior analysis results of the popular ransomware CryptoWall. Moreover, they present an application cooperating with the SDN controller and evaluate proof-of-concept implementations of the proposed mitigation system. The proposed methods rely on dynamic blacklisting of the ransomware proxy servers used to relay communication between the victim and the command and control server. In the first mitigation method, all DNS requests are forwarded to the controller by the application, then the controller checks domain names against the blacklist database. If a malicious domain is detected, the controller drops the packet, blocks all traffic from the source node, and alerts administrators. The second method is devised to increase the performance of the first method. The developed application sends a copy of DNS requests to the controller, and the controller and the application concurrently execute the same process defined in the first method. The authors claim that the proposed methods are feasible and efficient.

Ceron et al. [126] present an architecture named MARS (Malware Analysis Architecture) to dynamically reconfigure the network environment against malware actions using SDN. The presented architecture is composed of a sandbox, an SDN controller, and a resource pool. The sandbox executes the malware and performs connections to the Internet and/or to local network elements. Network traffic events are generated to the modules on the controller. The controller forwards or redirects intranet network flows to related devices in the resource pool, and manages flow characteristics (e.g., packet rate, throughput) to enforce predefined policies. The controller can also interact with the sandbox and restart a new analysis. The authors claim that the proposed solution can trigger more malware events than traditional solutions.

Hu et al. [127] propose Worm-Hunter, a closed-loop defense system against worms based on SDN. Worm-Hunter manages the flow tables of SDN switches and deploys different honeynet systems with different network structures dynamically. If Worm-Hunter detects anomalous traffic, it redirects suspected traffic into the honeynet to enable it to be analyzed safely. All of the attack behavior is recorded in the system for further analysis. Worm-Hunter is able to build multiple honeynet systems with the same topology.

Jin and Wang [128] categorize mobile malware attack types as follows: repackaging and updating attacks, drive-by download attacks, remote control, and information collection. They propose several mobile malware detection algorithms, and design and implement a malware detection system using SDN. The detection algorithms are implemented as modules inside the SDN controller using approaches like IP blacklist, connection success ratio, throttling connection, and aggregation analysis. The proposed system detects mobile malware by identifying suspicious network activities from real-time traffic using only connection establishment packets. After a connection is established, subsequent packets can go through the switch directly. The authors state that the proposed controller does not adversely affect network performance significantly.

Masoud et al. [129] express that web phishing attacks depend on the behavior of users but not on protocols. They propose a tailored SDN controller to tackle phishing attacks and implement a neural network based phishing prevention algorithm (PPA) on the controller. The proposed algorithm is tested using phishing test web pages in the home network. The authors state that PPA detects fake versions of web pages and enables access to the real versions of these pages. Chin et al. [130] propose another phishing detection and mitigation approach named PhishLimiter using an artificial neural network model to classify phishing attack signatures and apply real-time deep packet inspection. For each packet-in message that comes to the SDN controller, a phishing score is calculated and sent to the classifier. If the analysis determines malicious activity, packets are directed to a quarantine area. PhishLimiter defines two inspection modes: in fast mode, the packet is forwarded to the destination and a copy of it is stored for inspection, whereas in slow mode, all packets wait for the result of the inspection. The authors evaluated PhishLimiter using a real-world testbed environment and emails, and reported that PhishLimiter could detect and mitigate phishing attacks with an accuracy of 98.39%.

Table 8
SDN-based defense solutions against sniffer attacks.

Research	Defense type	Proposed methodology, Technique, or Procedure	Approaches	Deployment
DhcFlower [118]	Prevention	Random hopping end and route are applied in a period.	REROUTE-MPATH, ADDRESS-MOD	CNTL
[119]	Prevention	Packets in a flow are forwarded through different random routing paths periodically changed using SMT on a defined overlay network.	REROUTE-MPATH	CNTL
[120]	Prevention, Detection, Mitigation	Packets in a flow are forwarded through different random routing paths calculated by ant colony algorithm in a predefined period. Entropy matrix of network traffic characteristics is constructed using wavelet transform and principal component analysis to detect traffic anomaly.	REROUTE-MPATH	CNTL
[121]	Prevention	Stream encryption, spatial random scrambling, fragmentation, duplication, and shuffling packets inside the SDN switch.	PAYLOAD-ENC, REROUTE-MPATH, PAYLOAD-MOD	APP, SWCH
[122]	Prevention	Private protocol is created and standard protocol data are transformed to private protocol randomly. Receivers can remove the randomization to get standard protocols.	PAYLOAD-MOD	SWCH
[123]	Prevention	Each switch transmits only a portion of the packets exchanged from paths during communication.	REROUTE-MPATH	CNTL
[124]	Prevention	DNS requests are buffered for a few seconds and authentication packet is sent for each DNS request. If client responds to authentication packet, DNS request is allowed.	NAC	CNTL

Table 9
SDN-based defense solutions against malware, social engineering and web application attacks.

Research	Defense type	Proposed methodology, Technique, or Procedure	Approaches	Deployment
[125]	Detection, Mitigation	Check domains with the blacklist database in DNS requests. Block all traffic from infected host.	NETWORK-SEP, BLACKLIST	CNTL, APP
MARS [126]	Mitigation	Connection to internal devices are managed by SDN controller.	NETWORK-SEP	CNTL, APP
Worm-Hunter [127]	Detection, Mitigation	Signature matching detection and anomaly detection using the statistics. Suspicious traffic is directed and then observed into dynamic virtual honeynet.	SEND-TO-DECOY	APP
[128]	Detection, Mitigation	IP blacklist, connection success ratio, throttling and traffic aggregation are used to detect malware. Host is disconnected.	RATE-LIMIT, BLACKLIST	CNTL
[129]	Detection, Mitigation	Phishing prevention algorithm utilizes back propagation neural network model and denies to access phishing sites.	DROP	CNTL
PhishLimiter [130]	Detection, Mitigation	Controller manages scores for flows and application classifies flows using neural network model.	NETWORK-SEP, DROP	CNTL, APP
DBA [131]	Detection, Mitigation	Application decides DDoS attack and notifies DBA. Server address is changed dynamically.	ADDRESS-MOD	APP, SRVR
[132]	Detection, Mitigation	Detection sensor is deployed in front of protected applications, and detects anomaly using different algorithms. Traffic is redirected to provisioned honeypots.	SEND-TO-DECOY	APP

Lim et al. [131] propose a DDoS blocking application (DBA) to protect servers from botnet attacks. The scheme utilizes communication between the DBA running on the SDN controller and the server under attack, and orchestrates the defense through OpenFlow interfaces. The protected servers establish secure communication channels with DBA, and notify DBA about an attack. DBA then provides the redirected address to the server. Shtern et al. [132] propose a reference architecture that mitigates application-level low and slow DDoS (LSDDoS) attacks. The reference architecture is composed of LSDDoS detection sensor, automation controller, protected application, monitoring system, and shark tank. The protected application represents applications that will be defended against LSDDoS attacks. The detection sensor detects LSDDoS attacks and triggers the automation controller, which then provisions a restricted area named shark tank, and forwards malicious traffic to it. The monitoring system collects performance metrics from the components in the architecture. The authors also define performance model-based and off-the-shelf components based concrete architectures to show example alternatives.

Highlights: Table 9 provides a summary of the SDN-based anti-malware and anti-phishing systems reviewed in this subsection. The number of studies is small, but the variety in defensive approaches is relatively high. This is not surprising because it is not immediately obvious how SDN can help the defense against these application-layer threats, so proposals need to get creative.

Cyber threat intelligence (CTI) is a crucial weapon against malware, phishing, and web application attacks. Antivirus software and personal computer firewalls are not sufficient in this fight: they need to be supplemented by dynamic, automated defense mechanisms in the network. Integrating CTI into SDN is a promising wrinkle in that it enables both better prevention from and a quicker response to fast-growing threats on the web [133]. A more detailed discussion on CTI-assisted automated SDN defense is provided in Section 6.2.

5. Evaluation

The previous sections include extensive discussions on SDN-based proposals against several types of threats. In order to illustrate the variety in the literature, we present a taxonomy of SDN-based cyber defense solutions in Fig. 4. Classifying studies according to cyber threat category is one of the main features of our work, as stated earlier. As for defense type, we consider detection, prevention, and mitigation: A study can involve any combination of these three approaches, except for detection by itself due to our methodology explained in the beginning of Section 4. Each high-level defense strategy involves some combination of the underlying approaches in the taxonomy. The rest of this section presents various statistics related to the proposed taxonomy.

In this survey, 73 studies that propose SDN-based prevention and mitigation approaches against cyber threats are reviewed. Even though some of the mentioned studies apply or test their proposed approaches for multiple cyber threat categories, all studies are reviewed in one main cyber threat category in Section 4. Fig. 5(a) illustrates the percentage of studies related to different cyber threat categories. 79% of all 73 studies focus on mainly scanning, spoofing, or DoS attacks. Only 21% of studies have proposed approaches against sniffing, malware, social engineering, and web application attacks. Defense types of the solutions proposed in the studies are shown in Fig. 5(b). There are 27 studies which propose preventive defense approaches (11 for scanning, 8 for spoofing, 6 for sniffing, 2 for DoS attacks). Prevention approaches are common against scanning, spoofing, and sniffing attacks. 38 studies propose mitigation approaches for defense against cyber threats. Mitigation is the most commonly used defense type against DoS, malware, social engineering, and web application attacks. Besides, 8 studies develop both prevention and mitigation methods against scanning, spoofing, sniffing, and DoS attacks.

Fig. 6 shows the prevalence of each defense strategy grouped by attack type. The mainstream defense strategy for DoS, spoofing, and web application attacks is the policy-based strategy. The moving target defense strategy is most commonly used against spoofing and sniffing attacks. Some studies proposed to counter DoS and web application attacks use the machine learning based strategy to classify network traffic. Finally, the number of studies using the collaborative and distributed strategy to defend against attacks has increased in recent years.

There are different types of SDN simulation platforms, switch implementations, and test network environments used in the literature [134]. As a summary, the deployment-related taxonomy of SDN-based cyber defense solutions reviewed in our study is shown in Fig. 7.

SDN-based defense solutions are most commonly deployed on the controller, but deployment using an application and on the switches are also fairly common, as shown in Fig. 8(a). Defense approaches in the literature have been implemented using different SDN controllers (Fig. 8(b)). 16 studies have given no information about the SDN controller used (22%). Despite that, nearly half of the studies (41%) have used the POX controller [135] to implement their proposals. Besides, NOX [136], Ryu [137], and Floodlight [116] are the other most preferred controllers. Proposed defense approaches are generally evaluated on one or more test networks with SDN switches. The switches and the test networks used in the studies are given in Fig. 8(c) and Fig. 8(d), respectively. Half of the studies (36 studies) have used Mininet [138] to test their proposed defense approaches. 19 studies have created predefined test networks with physical machines and 16 studies have set up predefined test laboratories with virtual machines. Only two studies have tested their approaches in campus networks. Most commonly, Open vSwitch [139] (in 36 studies) has been used as the SDN switch in the studies. OpenFlow-enabled physical switches have been deployed to evaluate proposed defense approaches in 14 studies. Random topologies have been generated to evaluate the performance of proposed approaches in some studies. The most used random topology generation models are Erdos–Renyi, Waxman, Barabasi–Albert, and Watts–Strogatz models.

Table 10
Deployment locations of defense approaches.

Defense approach	CNTL	APP	SWCH	HOST	SRVR	APP-SWCH	Total
DROP	26	10	6			3	**45**
BLACKLIST	5	4	2			2	**13**
NETWORK-SEP	4	4					**8**
NAC	15	3	3				**21**
RATE-LIMIT	6	1	4			1	**12**
REROUTE-FLOW	4	2	2			1	**9**
REROUTE-MPATH	4	1	1				**6**
RANDOM-DECOY	4	3					**7**
PAYLOAD-ENC	1						**1**
VIRTUAL-VIEW	4	3					**7**
HEADER-MOD	3	3	1				**7**
ADDRESS-MOD	8	4	2	2	3	1	**20**
PAYLOAD-MOD	3	2	2				**7**
SEND-TO-DECOY	3	3	2				**8**
	90	**43**	**25**	**2**	**3**	**8**	

5.1. Evaluation of defense approach deployments

Most of the reviewed studies have introduced new components or proposed modifying the current SDN architecture components for their defense approaches. To implement some defense approaches, one or more components can be deployed onto the SDN architecture. The number of studies for each deployment location (illustrated in Fig. 3) is given in Fig. 9, classified by threat category. Furthermore, the numbers of deployment locations by defense approach are given in Table 10. *CNTL* and *SWCH* deployments are mainstream for the reviewed studies. Based on Fig. 9 and Table 10, review results about defense approach deployments are listed below:

- *CNTL deployments*: The most common deployment location to tackle scanning, spoofing, sniffing, and DoS attacks is the SDN controller extension (*CNTL*). *CNTL* deployment is proposed for nearly all defense approaches, but mostly used with *ADDRESS-MOD*, *DROP*, *VIRTUAL-ENV*, *REROUTE-FLOW*, *REROUTE-MPATH*, and *RANDOM-DECOY*.
- *APP deployments*: Custom SDN application deployment (*APP*) is used for all cyber threat types and nearly all defense approaches.
- *SWCH deployments*: Modification of default SDN switch behavior or management of rules have been proposed in some studies. SDN switch modification deployment (*SWCH*) is most applicable for scanning, sniffing, and DoS attacks. This deployment type is essential for *DROP* and *NAC* defense approaches.
- *HOST and SRVR deployments*: Agent installations are needed in some studies to defend against scanning and web application attacks. *HOST and SRVR* deployments have been proposed only for the *ADDRESS-MOD* approach.
- *APP-SWCH deployments*: Some studies propose an SDN application that communicates with SDN switches directly to deal with DoS attacks. Also, this deployment type has been suggested for *DROP* and *NAC* defense approaches.

5.2. Evaluation of defense approaches

In this part, studies from the literature have been categorized according to defense approaches listed in Table 4. The numbers for defense approaches by cyber threat category are given in Table 11. Most studies focus on DoS, spoofing, scanning attacks, and propose *DROP*, *NAC*, *ADDRESS-MOD*, and *REROUTE-FLOW* approaches to defend against these attacks.

Review results of defense approaches for different threat categories are summarized below:

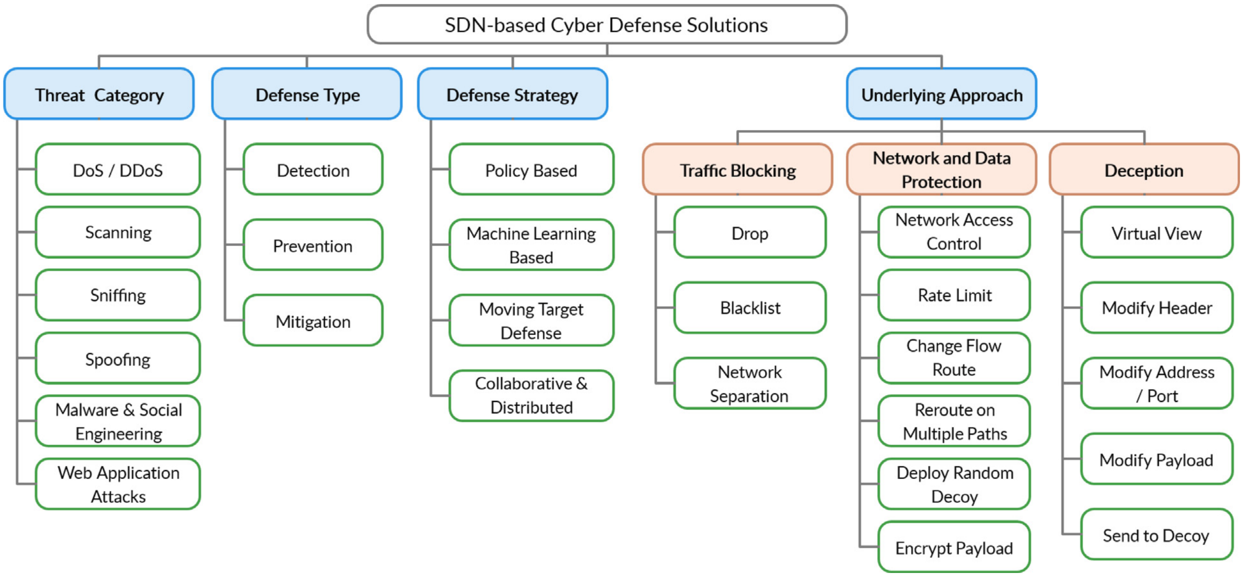
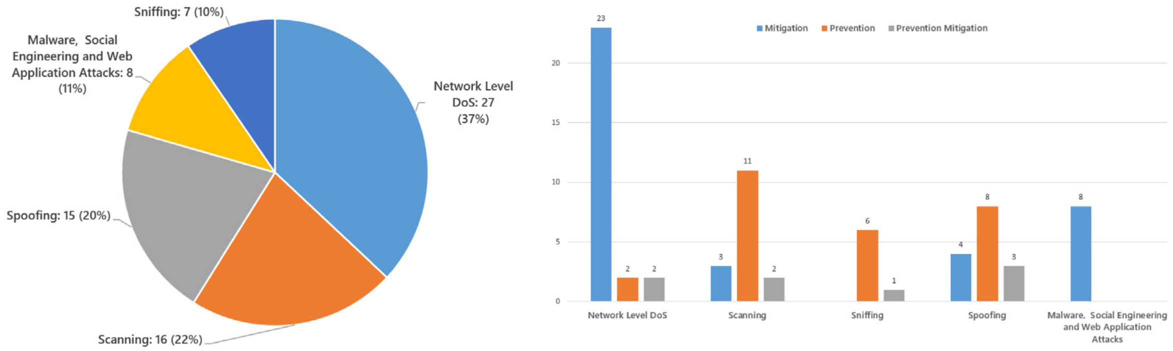


Fig. 4. Taxonomy of SDN-based cyber defense solutions.



(a) Percentages of studies by threat category

(b) Number of studies for defense type by threat category

Fig. 5. Numbers and percentages of studies.

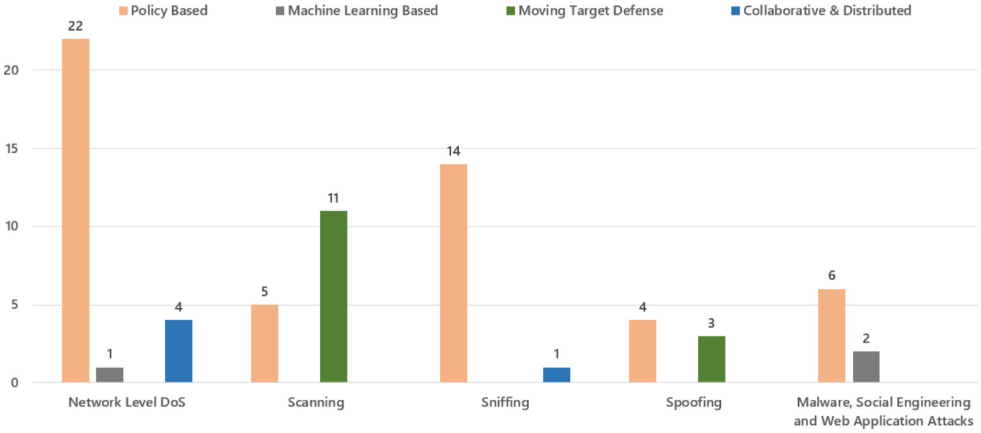


Fig. 6. Defense strategies by threat category.

- *Scanning attacks*: *ADDRESS-MOD*, *VIRTUAL-VIEW*, and *RANDOM-DECOY* are the main defense approaches against scanning attacks.
- *Spoofing attacks*: *DROP* and *NAC* defense approaches are usually applied against spoofing attacks. Additionally, *REROUTE-FLOW* and *HEADER-MOD* defense approaches are proposed in one study each to tackle spoofing attacks.
- *DoS attacks*: *DROP* and *REROUTE-FLOW* are the most commonly used approaches against DoS attacks. Besides, each of *BLACKLIST*, *NAC*, *RATE-LIMIT*, *HEADER-MOD*, *SEND-TO-DECOY* defense approaches is referenced in at least one study.

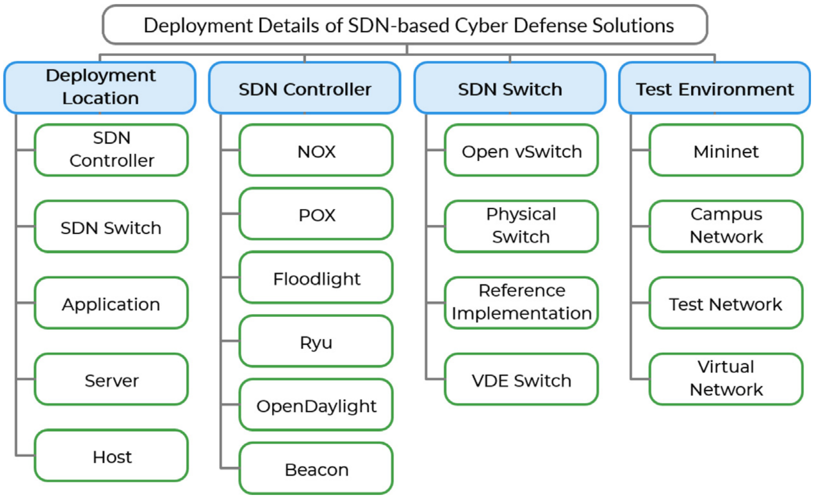
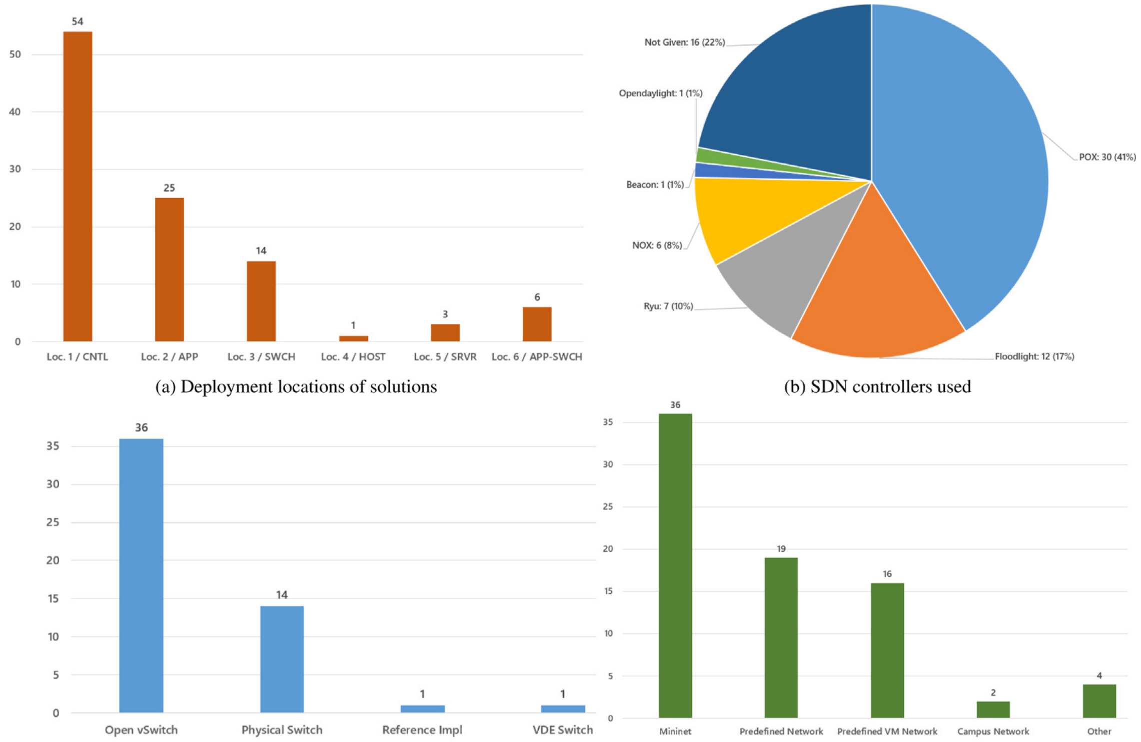


Fig. 7. Deployment details of SDN-based cyber defense solutions.



(a) Deployment locations of solutions

(b) SDN controllers used

(c) SDN switches used

(d) Test network infrastructure used

Fig. 8. Statistics about the deployment of SDN-based defense solutions.

- *Sniffing attacks*: *NAC*, *PAYLOAD-ENC*, *ADDRESS-MOD*, and *PAYLOAD-MOD* defense approaches can be used against sniffing attacks, but the most important preventive defense approach for sniffing attacks is *REROUTE-MPATH*.
- *Malware and social engineering attacks*: *DROP*, *NETWORK-SEP*, *RATE-LIMIT*, and *SEND-TO-DECOY* defense approaches are encountered for malware and social engineering attacks in the literature.
- *Web application attacks*: *ADDRESS-MOD* and *SEND-TO-DECOY* defense approaches are used against web application attacks.

DROP defense approach is mostly used against spoofing and DoS attacks, but it is also applicable for scanning, malware, and social engineering attacks. If traffic from trusted sources is accepted and any unknown traffic is blocked, this approach can be called *whitelisting*. The whitelist must be maintained by the SDN controller or an application. On the other hand, this approach must be used with other approaches for a more complete defensive strategy, for instance, *NAC* and *DROP* are used together against spoofing attacks. Besides, using *REROUTE-FLOW* and *DROP* together is generally applicable for DoS attacks. *BLACKLIST* defense approach is the complement of the whitelisting approach. A cyber threat list can be generated from known cyber threat intelligence sources and converted into a blacklist, which must be maintained by the SDN controller or an application. Threats in the blacklist are converted to drop or reroute to honeypot flow rules and sent to SDN switches. This approach is applicable for DoS, malware, social engineering, and web application attacks.

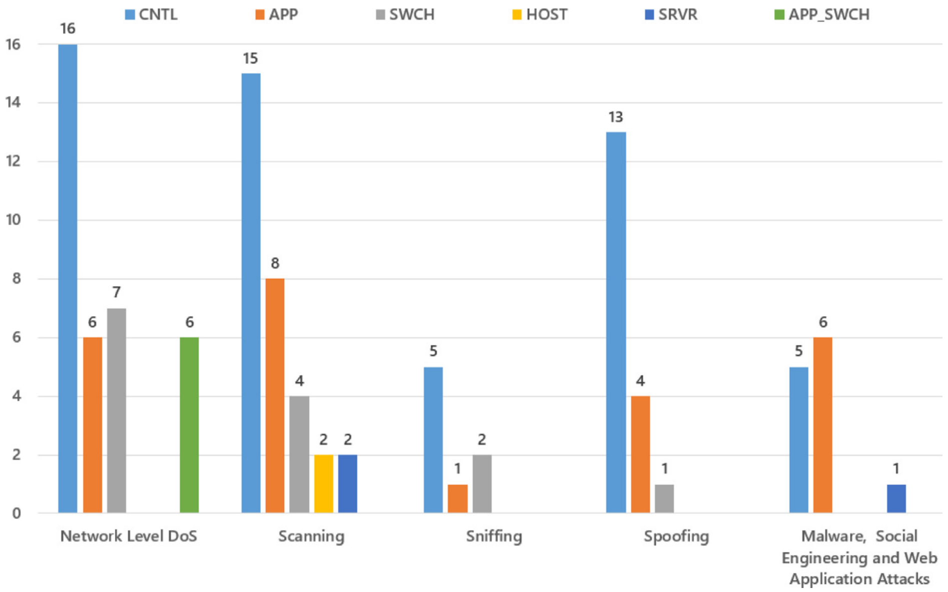


Fig. 9. Number of studies by deployment location of solutions.

Table 11
Defense approaches by threat category.

Defense approach	DoS	Scanning	Sniffing	Spoofing	Malware, Soc. Eng., Web App.	Total
DROP	20	2		14	2	**38**
BLACKLIST	6				3	**9**
NETWORK-SEP		1			3	**4**
NAC	2	1	1	13		**17**
RATE-LIMIT	7				1	**8**
REROUTE-FLOW	6	1		1		**8**
REROUTE-MPATH			5			**5**
RANDOM-DECOY		4				**4**
PAYLOAD-ENC			1			**1**
VIRTUAL-VIEW		4				**4**
HEADER-MOD	3	1		1		**5**
ADDRESS-MOD	1	6	1		2	**10**
PAYLOAD-MOD		2	3			**5**
SEND-TO-DECOY	2	2			2	**6**
	47	**24**	**11**	**29**	**13**	

However, a straightforward approach to blacklisting via flow table entries on SDN switches may not be applicable in cases where the blacklist includes more than a few thousand threat definitions. This is because the flow tables of SDN switches have limited capacity and typically only thousands of flow rules can be installed into the flow tables. On the other hand, millions of blacklist threat items may exist, so only threats with higher priority should be handled using the flow tables of SDN switches.

NETWORK-SEP defense approach is applicable against scanning, malware, and social engineering attacks. It is commonly used to inspect anomalies regarding traffic source, to isolate a portion of the network and to enforce organizational policies. This approach is applied with the *NAC* approach when a host tries to connect to the network, and with *SEND-TO-DECOY* to analyze the behavior of host traffic, for which a honeypot is typically deployed. Packets do not get dropped with this approach, instead, a tag is attached to each packet of a flow to identify and share information among network components. Tagged packets can traverse only allowed network hosts after relevant rules are installed on SDN switches. All network components (switches, routers, etc.) must know how to extract the tag and how to handle a tagged packet.

NAC defense approach is mainly used against spoofing attacks, and also applicable against scanning, DoS, and sniffing attacks. Legitimate network components (hosts, network devices, etc.) and access rights must be maintained by the SDN controller, which acts as an authentication and authorization server handling the first request. Legitimate network components can be verified using the SDN network topology, or an authentication protocol can be initiated. Unknown network components are not allowed to join the network and any unknown traffic is blocked in the network. Network access control can be applied at different levels (source MAC address, IP address, traffic path (source to destination) or high level protocol (HTTP, SSH, etc.)). While using an authentication protocol, the SDN controller or an application buffers packets, and only authenticated packets are allowed in the buffer. If an authentication protocol is activated, this approach is not transparent from hosts as each client must know the authentication mechanism and high performance buffering techniques are needed.

RATE-LIMIT approach is used against DoS, malware, and social engineering attacks. The SDN controller limits traffic or flow transmission rate to prevent starvation, or extends resources (e.g., virtual machines, servers) to comply with quality of service requirements using cloud computing capabilities. In this approach, traffic statistics must be monitored and checked against threshold values. Separate thresholds must be maintained for sources, paths, flows, targets, etc. *REROUTE-FLOW*, *BLACKLIST*, and *DROP* approaches can be applied to prevent malicious traffic along with *RATE-LIMIT*. On the other hand, traffic statistics must be collected periodically from SDN switches as burst attacks in short time periods cannot be detected from aggregated switch traffic statistics.

RANDOM-DECOY defense approach is mainly used against scanning attacks. A decoy can be a honeypot, network, or simulated host. Random IP addresses can be assigned to any decoy and traffic can be monitored for scanning attacks. This approach can be used with *VIRTUAL-VIEW* where each host has a different virtual network view, which is transparent from hosts and can be applied to selected hosts. Network views can be changed when hosts are connected to the network each time. Malicious network scans are significantly delayed using wide range subnet address spaces. Additionally, modified DNS and DHCP servers are used or the SDN controller operates as Gateway, DNS and DHCP server to maintain different network views. A virtual network view database must be maintained and shared among SDN controllers. Protocol behaviors (e.g., IP, ICMP, ARP) and packet transmission latency must be handled for each network packet according to the assigned virtual network view. However, these solutions are not fully applicable for hybrid networks due to the static network configurations used in hosts, and network traffic sniffing can be used to learn real network configurations. Further, relationships between generated virtual network views can be used to discover the actual network configuration.

Packet modification is another widely used defense approach. IP address modification (*ADDRESS-MOD*) defense approach is employed against scanning, sniffing, and web application attacks. This approach is transparent from hosts as the SDN controller maintains virtual IP addresses and changes them periodically. A virtual IP address mapping database must be maintained and shared between SDN controllers. A custom DNS server serves the hosts to handle the resolution of names to virtual IP addresses instead of the real ones. The downside of this approach is that the process of obtaining virtual IPs may cause high overhead in the network. Also, this approach may not able to dynamically detect the source of scanning traffic. Another packet modification approach is header modification (*HEADER-MOD*), which is used against scanning, spoofing, and DoS attacks. It is mostly used to prevent fingerprinting attacks and to assign tags to packets. For sniffing attacks, payload encryption (*PAYLOAD-ENC*) approach can be used with an added overhead. Another defense approach against sniffing attacks, especially fingerprinting attacks, is payload modification *PAYLOAD-MOD*. With *PAYLOAD-MOD* or *HEADER-MOD* approaches, ICMP, TCP, and application-level protocols must be processed, and operating system and protocol behaviors must be analyzed and simulated.

6. Open research issues and future directions

The number and variety of studies reviewed in this paper indicate that using SDN to develop cyber defense solutions is a popular research area. Prevention and mitigation approaches using SDN have various challenges and lead to new research questions which have not been answered completely in the literature. Some of these high-level research questions are:

- How can high performance, scalable, high availability, customizable, easier to deploy, practical, and secure SDN-based defense solutions be implemented?
- How can SDN-based defense solutions be applied in hybrid networks?
- How should defense approaches be selected and applied for cyber threats adaptively?
- How should defense approaches be evaluated and compared with other solutions?
- How can SDN-based defense approaches be used with new technologies such as 5G and IoT?
- How can SDN make future technologies more secure?

The studies reviewed in this paper have certain shortcomings that should be addressed in future works. The most significant issues in the current literature are listed below:

- Most studies rely on the continued operation of a single controller, which may not be possible due to the many types of attacks (flooding, hijacking, etc.) targeting the controller.
- Most studies use small topologies and/or simplistic scenarios to evaluate the proposed defense mechanisms. This brings into question the applicability of these methods in real large-scale networks.
- Some proposals require changes to SDN infrastructure elements such as switches and other network appliances. These solutions are unlikely to be used in practice as any requirement for major modification in the infrastructure is a barrier to deployment.
- Most proposals assume that the entire network is an SDN infrastructure, and there is a lack of studies considering hybrid networks where SDN and traditional networking are used together.
- There is a lack of solutions utilizing service function chaining (SFC) and examining the use of new protocols such as Network Service Header (NSH) for building defense mechanisms.

In the following subsections, we define and list some challenges which need be investigated, and suggest directions for further research, based on the questions and issues mentioned above.

6.1. More secure, reliable, and scalable SDN architectures

Defending SDN infrastructure components and offsetting the new risks created by SDN require carefully designed architectures for both the network and the software to control it. Security, reliability, and scalability are three indispensable goals to be met in the design of network and controller architectures for SDN.

For more scalable and reliable architectures, a controller hierarchy or multiple coordinated controllers could potentially be utilized. These controllers could work together to realize extensible network defense approaches. Developing such solutions that manage the responsibilities of multiple controllers working in concert is an important research question. In a relevant study, Hu et al. [140] propose an SDN controller architecture called Mimic Network Operating System (MNOS) based on the idea of cyber mimic defense, which improves resiliency and security through increased redundancy. More specifically, MNOS employs multiple different controllers and introduces a mimic layer into the SDN controller architecture to select a subset of these controllers to be active together at a given time. This architecture enhances security against attacks attempting to exploit vulnerabilities in the SDN controller to achieve results such as controller hijacking and data modification. The heterogeneity in the set of controllers reduces the risk that a vulnerability in a particular controller may become an attack vector.

Any multi-controller architecture such as the one described above will need to employ new modules to coordinate the actions of the controllers and maintain state consistency at all times. These new modules should be developed in such a way to minimize the risk of causing security vulnerabilities. Additionally, the controllers should be close enough in functionality to be used as substitutes for one another, but also dissimilar enough to ensure that the same vulnerability will not exist in two different controllers. For this reason, another important challenge is devising methodologies for evaluating both the functional similarity and the structural dissimilarity between two given controllers, and finding the best way of choosing a set of diverse but functionally equivalent controllers.

Dynamic, distributed, highly available, and scalable SDN-based network defense architectures will make use of network service functions. The management and orchestration of security functions is a popular research area, and there is a need for increased focus on this problem in emerging domains such as 5G, IoT, and Tactile Internet. Specifically, more studies are needed to analyze the potential impact of service function chaining (SFC) and protocols like Network Service Header (NSH) on the performance of network defense solutions.

6.2. Seamless integration of SDN with other technologies to create added value

Integrating SDN-enabled networks with cyber threat intelligence (CTI) services offers exciting opportunities for security. CTI typically contains various information about cyber threats that may include threat type, indicators, threat actor, tactics, procedures, countermeasures, etc. Organizations obtaining this information can task their security teams with interpreting it and incorporating it into their defense mechanisms, but this approach may not be agile enough to take timely actions. Instead, the data collected from CTI sources can be used to automatically generate forwarding rules in SDN for quicker response to cyber threats. However, not every piece of CTI data can easily be turned into a flow table entry on an SDN switch: there needs to be a preprocessing stage to determine whether a particular piece of intelligence can be used to construct a flow rule. To this end, classifying, enriching, and prioritizing CTI data via machine learning will enable a stronger and more efficient network defense system.

Softwarization brings about new challenges and risks for network security, such as widened attack surface and increased potential impact for attacks. Novel proactive defense mechanisms need to be developed to counter growing risks for SDN and network functions virtualization (NFV) based cloud and carrier networks. A particularly important technology that will be widely used with 5G and future generation telecommunication technologies is *network slicing*, which necessitates a meticulous approach to security to achieve its requirements of strong isolation among slices and performance guarantees in the presence of attacks. We expect that network slicing security in 5G and beyond will be a major research topic for the next several years.

Blockchain is another emerging technology with a wide range of application that includes cyber security. It has introduced decentralized, transparent, robust, auditable, and secure architecture solutions for various business domains. Security researchers have proposed SDN and blockchain based security architectures to realize distributed security mechanisms to deal with cyber attacks. In the security literature, there are two main research areas bringing together SDN and blockchain: blockchain-based security enhancements for SDN-enabled networks, and blockchain-based distributed DDoS mitigation using SDN. Blockchain-based security enhancements for SDN-enabled networks have been proposed in technological domains such as IoT [141–143], 5G [144, 145], SDON [146], etc. In the area of blockchain-based distributed DDoS mitigation, there are a few recent works [109–111] using SDN. We should note that neither blockchain nor SDN alone is sufficient to mitigate DDoS attacks, however, recent studies focus on the potential of blockchain to help reduce the resource consumption and improve the performance of mitigation systems. In addition to these, blockchains can be used to share cyber threat intelligence between multiple domains and to mitigate other cyber threats like ransomware, phishing, etc. More research needs to be conducted in all of these areas to fully realize the potential of blockchain-SDN integration.

IoT has become a particularly active domain for blockchain applications in recent years. Ali et al. [147] describe general application areas for IoT security through blockchains and Pohrmen et al. [148] give brief information about blockchain-based SDN-IoT architectures and frameworks. Blockchain-based IoT solutions have inherent resource constraints, so there are research opportunities in developing optimized and lightweight blockchain-based SDN security mechanisms for IoT [147,148]. One way to simplify the implementation could be removing some features from blockchain (e.g., security features against double spending) which are not necessary for network security solutions. Moreover, most blockchain solutions for IoT use smart contracts, however, security standards for writing secure contracts are not defined and used in IoT solutions. Thus, the issue of developing IoT smart contracts using SDN in such a way that will not introduce new vulnerabilities is also a potential avenue for future work. Another research topic is distributed SDN control for IoT: blockchains can be integrated into the SDN control plane to decentralize and enhance security [147]. Different blockchain-based architectures can be defined to manage IoT networks, govern resources and defend them from attacks, such as the one proposed by Yang et al. [142]. In 5G, blockchain technology can provide trusted node collaboration in networks, so multiple network domains can communicate securely using blockchains [149]. Yang et al. [144] present an example of blockchain-based trusted multi-domain collaboration systems in 5G. Investigating such uses of blockchain for trusted multi-domain or network communication is another promising research direction.

6.3. New SDN-based defense approaches

There are several SDN-based defense primitives discussed throughout this study. Both new primitives and modular defense systems making use of multiple primitives are necessary to deal with modern cyber threats. In particular, researchers should develop enhanced methods to improve the security of common network protocols (ARP, DHCP, DNS, etc.) as well as newer protocols such as NDP in IPv6. Defense mechanisms must be adaptive and consider threat categories, attacker capabilities, and available network resources. For instance, the adversarial network forensics approach mentioned at the end of Section 4.1 presents a new challenge to security researchers: designing networks to be resistant to attack methods which exploit the openness of the SDN architecture.

More studies are needed on SDN-based attack deception and prevention using intelligent multipath routing, especially in hybrid networks where SDN has been only partially deployed. Proposed multipath routing solutions should be able to overcome the complications arising from the differences among the capabilities of data plane elements. In addition, best practices should be developed regarding the response to various cyber threats by defining network service function chains based on threat category and applying these chains effectively. We believe this survey provides useful information on how different SDN-based defense primitives have been employed to counter each type of attack. Hence, researchers can use this work in their effort to construct a blueprint for combining different defensive actions into an effective and adaptable security policy.

6.4. Practical evaluation methodologies

Throughout this study, we have observed that there are nearly as many different approaches to evaluating defense methods as there are publications in this area. For example, test topologies used in different works include toy networks with only a few hosts, larger topologies constructed on tools like Mininet, campus networks, and large-scale testbeds. Also, some studies consider only a single or very few attackers while others examine the behavior of their solutions against coordinated, distributed attacks. Last but not least, there is considerable variability in the evaluation metrics of different studies: for instance, some papers proposing solutions against DoS focus on the utilization of resources such as CPU and flow table space as their primary metrics, while others highlight end-to-end effects like data transfer rates and round-trip time. These discrepancies make it harder to compare the merits of different solutions in the literature.

In the field of moving target defense (MTD), a recent work by Connell et al. [150] recognizes the aforementioned problem and tackles it by proposing an analytic model to evaluate MTD solutions. The proposed model enables computing metrics such as response time, resource availability, the chance of success for the attacker, etc. for a given MTD system. It also allows exploring the tradeoff between security and performance by varying several parameters including reconfiguration rate, request arrival rate, time taken to reconfigure a resource, and maximum number of simultaneous reconfigurations. The broader security research community would benefit from similar methodologies and benchmarks for evaluating all classes of SDN defense approaches based on performance, scalability, availability, applicability, and security.

The amount, quality, and variety of network traffic data are also important for evaluating the proposed defense methods. For this reason, traffic data generated and collected on realistic SDN-based network topologies need to be compiled into various datasets for different kinds of attacks and made available to researchers. This is particularly important for both developing and evaluating machine learning based solutions to attack detection and response in SDN.

7. Conclusion

SDN has emerged as a new network architecture with numerous benefits for managing networks dynamically. Along with NFV, these two popular paradigms have ushered in the age of network softwarization, which offers opportunities related to security for both defenders and attackers. Understanding cyber threats and using cyber threat intelligence data to secure software networks are now more critical for effective defense against both known threats and zero day attacks.

This study presents a systematic review of cyber threat categories and related defense approaches using SDN in the literature. Our focus is on SDN-based solutions for preventing or mitigating attacks categorized as scanning, spoofing, DoS, sniffing, malware and social engineering, and web application attacks. For each category, the studies in the literature are reviewed and the defense approach in each one is summarized. Furthermore, a thorough evaluation of defense approaches, as well as open issues and future directions are given. We believe this study will be valuable for researchers in this field and security professionals looking to apply new defense solutions in modern networking environments.

CRediT authorship contribution statement

Ozgur Yurekten: Conceptualization, Methodology, Investigation, Writing - original draft, Visualization. **Mehmet Demirci:** Writing - review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] McAfeeLabs, Threats Report, August 2019, Tech. Rep., McAfee Labs, 2019, p. 35, URL http://web.archive.org/web/20200521150107/https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-aug-2019.pdf, (Accessed: 22 May 2020).

[2] Kaspersky, Kaspersky Security Bulletin 2019, Statistics, Tech. Rep., Kaspersky Lab, 2019, p. 6, URL https://securelist.com/kaspersky-security-bulletin-2019-statistics/95475/, (Accessed: 22 May 2020).

[3] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, DDoS in the IoT: Mirai and other Botnets, IEEE Comput. 50 (7) (2017) 80–84, http://dx.doi.org/10.1109/MC.2017.201.

[4] R. Masoudi, A. Ghaffari, Software defined networks: A survey, J. Netw. Comput. Appl. 67 (2016) 1–25, http://dx.doi.org/10.1016/j.jnca.2016.03.016.

[5] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76, http://dx.doi.org/10.1109/JPROC.2014.2371999.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: Enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69, http://dx.doi.org/10.1145/1355734.1355746.

[7] S. Qamar, Z. Anwar, M.A. Rahman, E. Al-Shaer, B.T. Chu, Data-driven analytics for cyber-threat intelligence and information sharing, Comput. Secur. 67 (2017) 35–58, http://dx.doi.org/10.1016/j.cose.2017.02.005.

[8] STIX 2.1 specification, 2020, https://www.oasis-open.org/standards/#stix2.1, (Accessed: 22 May 2020).

[9] RFC 5070 - IODEF (the incident object description exchange format), 2014, https://tools.ietf.org/html/rfc7203, (Accessed: 22 May 2020).

[10] OpenIoC, 2019, https://github.com/mandiant/OpenIOC_1.1, (Accessed: 22 May 2020).

[11] M.C. Dacier, H. König, R. Cwalinski, F. Kargl, S. Dietrich, Security challenges and opportunities of software-defined networking, IEEE Secur. Priv. 15 (2) (2017) 96–100, http://dx.doi.org/10.1109/MSP.2017.46.

[12] M.T. Kurniawan, S. Yazid, A systematic literature review of security software defined network: Research trends, threat, attack, detect, mitigate, and countermeasure, in: Proceedings of the 3rd International Conference on Telecommunications and Communication Engineering, Association for Computing Machinery, 2019, pp. 39–45, http://dx.doi.org/10.1145/3369555.3369567.

[13] I. Alsmadi, D. Xu, Security of software defined networks: A survey, Comput. Secur. 53 (2015) 79–108, http://dx.doi.org/10.1016/j.cose.2015.05.006.

[14] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, Security in software defined networks: A survey, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2317–2346, http://dx.doi.org/10.1109/COMST.2015.2474118.

[15] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Commun. Surv. Tutor. 18 (1) (2016) 623–654, http://dx.doi.org/10.1109/COMST.2015.2453114.

[16] S. Achleitner, T.L. Porta, P. McDaniel, S. Sugrim, S.V. Krishnamurthy, R. Chadha, Cyber deception: Virtual networks to defend insider reconnaissance, in: 8th ACM CCS International Workshop on Managing Insider Security Threats, MIST 2016, 2016, pp. 57–68, http://dx.doi.org/10.1145/2995959.2995962.

[17] A. Shaghaghi, M.A. Kaafar, R. Buyya, S. Jha, Software-defined network (SDN) data plane security: Issues, solutions, and future directions, in: Handbook of Computer Networks and Cyber Security, Springer, 2020, pp. 341–387, http://dx.doi.org/10.1007/978-3-030-22277-2_14.

[18] D.B. Rawat, S.R. Reddy, Software defined networking architecture, security and energy efficiency: A survey, IEEE Commun. Surv. Tutor. 19 (1) (2017) 325–346, http://dx.doi.org/10.1109/COMST.2016.2618874.

[19] J.C.C. Chica, J.C. Imbachi, J.F. Botero, Security in SDN: A comprehensive survey, J. Netw. Comput. Appl. 159 (102595) (2020) 1–23, http://dx.doi.org/10.1016/j.jnca.2020.102595.

[20] I. Farris, T. Taleb, Y. Khettab, J. Song, A survey on emerging SDN and NFV security mechanisms for IoT systems, IEEE Commun. Surv. Tutor. 21 (1) (2019) 812–837, http://dx.doi.org/10.1109/COMST.2018.2862350.

[21] R. Swami, M. Dave, V. Ranga, Software-defined networking-based DDoS defense mechanisms, ACM Comput. Surv. 52 (2) (2019) 1–36, http://dx.doi.org/10.1145/3301614.

[22] S.E. Jasper, U.S. cyber threat intelligence sharing frameworks, Int. J. Intell. Counter Intell. 30 (1) (2017) 53–65, http://dx.doi.org/10.1080/08850607.2016.1230701.

[23] TAXII v2.1 specification, 2020, https://www.oasis-open.org/standards/#taxii2.1, (Accessed: 22 May 2020).

[24] S. Barnum, R. Martin, B. Worrell, I. Kirillov, The CybOX Language Specification, Tech. Rep., The MITRE Corporation, 2012, (Accessed: 22 May 2020), URL https://cybox.mitre.org/language/specifications/CybOX_Language_Core_Specification_v1.0.pdf.

[25] D. Rhoades, Machine actionable indicators of compromise, in: International Carnahan Conference on Security Technology, ICCST2014, 2014, pp. 1–5, http://dx.doi.org/10.1109/CCST.2014.6987016.

[26] E. Asgarli, E. Burger, Semantic ontologies for cyber threat sharing standards, in: IEEE Symposium on Technologies for Homeland Security, HST 2016, 2016, pp. 1–6, http://dx.doi.org/10.1109/THS.2016.7568896.

[27] R. Mijumbi, J. Serrat, J.-l. Gorricho, N. Bouten, F.D. Turck, B. Raouf, Network function virtualization: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262, http://dx.doi.org/10.1109/COMST.2015.2477041.

[28] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, X. Zheng, SD-anti-DDoS: Fast and efficient DDoS defense in software-defined networks, J. Netw. Comput. Appl. 68 (2016) 65–79, http://dx.doi.org/10.1016/j.jnca.2016.04.005.

[29] Z. Shu, J. Wan, D. Li, J. Lin, A.V. Vasilakos, M. Imran, Security in software-defined networking: Threats and countermeasures, Mob. Netw. Appl. 21 (5) (2016) 764–776, http://dx.doi.org/10.1007/s11036-016-0676-x.

[30] S. Wang, J. Wu, W. Yang, L.-h. Guo, Novel architectures and security solutions of programmable software-defined networking: a comprehensive survey, Front. Inf. Technol. Electron. Eng. 19 (12) (2018) 1500–1521, http://dx.doi.org/10.1631/FITEE.1800575.

[31] T. Han, S.R.U. Jan, Z. Tan, M. Usman, M.A. Jan, R. Khan, Y. Xu, A comprehensive survey of security threats and their mitigation techniques for next-generation SDN controllers, Concurr. Comput.: Pract. Exper. (2019) e5300, http://dx.doi.org/10.1002/cpe.5300.

[32] J. Benabbou, K. Elbaamrani, N. Idboufker, Security in OpenFlow-based SDN, opportunities and challenges, Photonic Netw. Commun. 37 (1) (2019) 1–23, http://dx.doi.org/10.1007/s11107-018-0803-7.

[33] K. Thimmaraju, B. Shastry, T. Fiebig, F. Hetzelt, J.-P. Seifert, A. Feldmann, S. Schmid, Taking control of sdn-based cloud systems via the data plane, in: Proceedings of the Symposium on SDN Research, ACM, 2018, p. 1, http://dx.doi.org/10.1145/3185467.3185468.

[34] S. Achleitner, T. La Porta, T. Jaeger, P. McDaniel, Adversarial network forensics in software defined networking, in: Proceedings of the Symposium on SDN Research, ACM, 2017, pp. 8–20, http://dx.doi.org/10.1145/3050220.3050223.

[35] U. Ghosh, P. Chatterjee, S. Shetty, Securing SDN-enabled smart power grids: SDN-enabled smart grid security, in: Cyber-Physical Systems for Next-Generation Networks, IGI Global, 2018, pp. 79–98, http://dx.doi.org/10.4018/978-1-5225-5510-0.ch004.

[36] S. Demirci, S. Sagiroglu, Software-defined networking for improving security in smart grid systems, in: 2018 7th International Conference on Renewable Energy Research and Applications, ICRERA, IEEE, 2018, pp. 1021–1026, http://dx.doi.org/10.1109/ICRERA.2018.8567005.

[37] N.E. Petroulakis, K. Fysarakis, I. Askoxylakis, G. Spanoudakis, Reactive security for SDN/NFV-enabled industrial networks leveraging service function chaining, Trans. Emerg. Telecommun. Technol. 29 (7) (2018) e3269, http://dx.doi.org/10.1002/ett.3269.

[38] STIX 1.2.1 specification, 2016, https://www.oasis-open.org/standards/#stix1.2.1, (Accessed: 22 May 2020).

[39] A. Sethi, S. Barnum, Introduction to attack patterns, 2013, https://www.us-cert.gov/bsi/articles/knowledge/attack-patterns/introduction-to-attack-patterns, (Accessed: 22 May 2020).

[40] Common attack pattern enumeration and classification (CAPEC), 2017, https://capec.mitre.org, (Accessed: 22 May 2020).

[41] MITRE adversarial tactics, techniques, and common knowledge (MITRE ATT&CK), 2020, https://attack.mitre.org/, (Accessed: 22 May 2020).

[42] G. Canbek, S. Sagiroglu, N. Baykal, New comprehensive taxonomies on mobile security and malware analysis, Int. J. Inf. Secur. Sci. 5 (4) (2016) 106–138, URL https://www.ijiss.org/ijiss/index.php/ijiss/article/view/227/pdf_43.

[43] S. Hansman, R. Hunt, A taxonomy of network and computer attacks, Comput. Secur. 24 (2005) 31–43, http://dx.doi.org/10.1016/j.cose.2004.06.011.

[44] S. Jin, Y. Wang, X. Cui, X. Yun, A review of classification methods for network vulnerability, in: IEEE International Conference on Systems, Man, and Cybernetics, SMC 2009, 2009, pp. 1171–1175, http://dx.doi.org/10.1109/ICSMC.2009.5345951.

[45] C.B. Simmons, S.G. Shiva, H. Bedi, D. Dasgupta, AVOIDIT: A cyber attack taxonomy, in: 9th Annual Symposium on Information Assurance, ASIA 2014, 2014, pp. 2–12, (Accessed: 22 May 2020), https://www.albany.edu/iasymposium/proceedings/2014/ASIA14Proceedings.pdf.

[46] S.H. Amer, J.A. Hamilton Jr., Intrusion detection systems (IDS) taxonomy - a short review, Defensive Cyber Secur.: Process. Policies 13 (2) (2010) 22–30, (Accessed: 22 May 2020), https://www.csiac.org/wp-content/uploads/2016/02/2010_06_29_DefensiveCyberSecurity.pdf.

[47] D.L. Lough, A Taxonomy of Computer Attacks with Applications to Wireless Networks (Doctor of philosophy), Virginia Polytechnic Institute and State University, 2001, http://web.archive.org/web/20200522175500/https://vtechworks.lib.vt.edu/handle/10919/27242.

[48] NTT Security, Global Threat Intelligence Report, Tech. Rep., NTT Security, 2019, p. 7, (Accessed: 22 May 2020), URL https://www.nttsecurity.com/docs/librariesprovider3/resources/2019-gtir/2019_gtir_report_2019_uea_v2.pdf.

[49] L. Neely, SANS 2017 Threat Landscape Survey: Users on the Front Line, Tech. Rep., SANS Institute, 2017, pp. 5–6, (Accessed: 22 May 2020), URL https://www.qualys.com/forms/whitepapers/sans-2017-threat-landscape-survey-users-front-line.

[50] Symantec, Internet Security Threat Report, Tech. Rep., (22) Symantec, 2017, (Accessed: 22 May 2020), URL https://docs.broadcom.com/doc/istr-22-2017-en.

[51] Z. Wu, Y. Ou, Y. Liu, A taxonomy of network and computer attacks based on responses, in: International Conference of Information Technology, Computer Engineering and Management Sciences, 2011, pp. 26–29, http://dx.doi.org/10.1109/ICM.2011.363.

[52] M. Kjaerland, A taxonomy and comparison of computer security incidents from the commercial and government sectors, Comput. Secur. 25 (2006) 522–538, http://dx.doi.org/10.1016/j.cose.2006.08.004.

[53] A.V. Uzunov, E.B. Fernandez, An extensible pattern-based library and taxonomy of security threats for distributed systems, Comput. Stand. Interfaces 36 (4) (2014) 734–747, http://dx.doi.org/10.1016/j.csi.2013.12.008.

[54] N. Hoque, M.H. Bhuyan, R.C. Baishya, D.K. Bhattacharyya, J.K. Kalita, Network attacks: Taxonomy, tools and systems, J. Netw. Comput. Appl. 40 (2014) 307–324, http://dx.doi.org/10.1016/j.jnca.2013.08.001.

[55] M. Fu, W. Zhenxing, G. Yi, L. Zhang, A security threats taxonomy for routing system intrusion detection, in: 12th International Conference on Computational Intelligence and Security, CIS 2016, 2016, pp. 267–270, http://dx.doi.org/10.1109/CIS.2016.0068.

[56] McAfee, Threats Report, September 2017, Tech. Rep., McAfee Labs, 2017, pp. 60–65, (Accessed: 22 May 2020), URL https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-sept-2017.pdf.

[57] S. Achleitner, T.F.L. Porta, P. McDaniel, S. Sugrim, S.V. Krishnamurthy, R. Chadha, Deceiving network reconnaissance using SDN-based virtual topologies, IEEE Trans. Netw. Serv. Manag. 14 (4) (2017) 1098–1112, http://dx.doi.org/10.1109/TNSM.2017.2724239.

[58] C.-Y.J. Chiang, Y.M. Gottlieb, S.J. Sugrim, R. Chadha, C. Serban, A. Poylisher, L.M. Marvel, J. Santos, ACyDS: An adaptive cyber deception system, in: IEEE Military Communications Conference, MILCOM 2016, 2016, pp. 800–805, http://dx.doi.org/10.1109/MILCOM.2016.7795427.

[59] S. Robertson, S. Alexander, J. Micallef, J. Pucci, J. Tanis, A. Macera, CINDAM: Customized information networks for deception and attack mitigation, in: IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2015, 2015, pp. 114–119, http://dx.doi.org/10.1109/SASOW.2015.23.

[60] J.H. Jafarian, E. Al-Shaer, Q. Duan, Openflow random host mutation: Transparent moving target defense using software defined networking, in: First Workshop on Hot Topics in Software Defined Networks, HotSDN 2012, 2012, pp. 127–132, http://dx.doi.org/10.1145/2342441.2342467.

[61] J.H.H. Jafarian, E. Al-Shaer, Q. Duan, Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers, in: First ACM Workshop on Moving Target Defense, MTD 2014, 2014, pp. 69–78, http://dx.doi.org/10.1145/2663474.2663483.

[62] J.H. Jafarian, E. Al-Shaer, Q. Duan, An effective address mutation approach for disrupting reconnaissance attacks, IEEE Trans. Inf. Forensics Secur. 10 (12) (2015) 2562–2577, http://dx.doi.org/10.1109/TIFS.2015.2467358.

[63] J.H. Jafarian, E. Al-Shaer, Q. Duan, Adversary-aware IP address randomization for proactive agility against sophisticated attackers, in: IEEE Conference on Computer Communications, INFOCOM 2015, 2015, pp. 738–746, http://dx.doi.org/10.1109/INFOCOM.2015.7218443.

[64] D. Ma, C. Lei, L. Wang, H. Zhang, Z. Xu, M. Li, A self-adaptive hopping approach of moving target defense to thwart scanning attacks, in: Lecture Notes in Computer Science, Vol. 9977, Springer, 2016, pp. 39–53, http://dx.doi.org/10.1007/978-3-319-50011-9_4.

[65] D.C. Macfarland, C.A. Shue, The SDN shuffle: Creating a moving-target defense using host-based software-defined networking, in: 2nd ACM Workshop on Moving Target Defense, MTD 2015, 2015, pp. 37–41, http://dx.doi.org/10.1145/2808475.2808485.

[66] K. Wang, X. Chen, Y. Zhu, Random domain name and address mutation (RDAM) for thwarting reconnaissance attacks, PLoS One 12 (5) (2017) 1–22, http://dx.doi.org/10.1371/journal.pone.0177111.

[67] L. Wang, D. Wu, Moving target defense against network reconnaissance with software defined networking, Lecture Notes in Comput. Sci. 9866 (2016) 203–217, http://dx.doi.org/10.1007/978-3-319-45871-7_13.

[68] Z. Zhao, F. Liu, D. Gong, An SDN-based fingerprint hopping method to prevent fingerprinting attacks, Secur. Commun. Netw. 2017 (2017) 1–12, http://dx.doi.org/10.1155/2017/1560594.

[69] P. Kampanakis, H. Perros, T. Beyene, SDN-based solutions for moving target defense network protection, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2014, 2014, pp. 1–6, http://dx.doi.org/10.1109/WoWMoM.2014.6918979.

[70] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, M. Tyson, FRESCO: Modular composable security services for software-defined networks, in: Network and Distributed System Security Symposium, NDSS 2013, 2013, pp. 1–16, URL http://hdl.handle.net/10203/205914.

[71] K. Cabaj, M. Gregorczyk, W. Mazurczyk, P. Nowakowski, P. Zorawski, SDN-Based mitigation of scanning attacks for the 5g internet of radio light system, in: Proceedings of the 13th International Conference on Availability, Reliability and Security, Association for Computing Machinery, 2018, pp. 1–10, http://dx.doi.org/10.1145/3230833.3233248.

[72] N. Sahri, K. Okamura, Cauth-protecting DNS application from spoofing attacks, Int. J. Comput. Sci. Netw. Secur. 16 (6) (2016) 125–134, URL http://paper.ijcsns.org/07_book/201606/20160615.pdf.

[73] N. Sahri, K. Okamura, Protecting DNS services from IP spoofing: SDN collaborative authentication approach, in: 11th International Conference on Future Internet Technologies, CFI 2016, 2016, pp. 83–89, http://dx.doi.org/10.1145/2935663.2935666.

[74] N. Sahri, K. Okamura, Collaborative spoofing detection and mitigation-SDN based looping authentication for DNS services, in: IEEE 40th Annual Computer Software and Applications Conference, COMPSAC 2016, 2016, pp. 565–570, http://dx.doi.org/10.1109/COMPSAC.2016.6.

[75] M.Z. Masoud, Y. Jaradat, I. Jannoud, On preventing ARP poisoning attack utilizing software defined network (SDN) paradigm, in: IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2015, 2015, pp. 1–5, http://dx.doi.org/10.1109/AEECT.2015.7360549.

[76] J.H. Cox, R.J. Clark, H.L. Owen, Leveraging SDN for ARP security, in: SoutheastCon 2016, 2016, pp. 1–8, http://dx.doi.org/10.1109/SECON.2016.7506644.

[77] A. Nehra, M. Tripathi, M.S. Gaur, FICUR: Employing SDN programmability to secure ARP, in: IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017, 2017, pp. 1–8, http://dx.doi.org/10.1109/CCWC.2017.7868450.

[78] F. Ubaid, R. Amin, F.B. Ubaid, M.M. Iqbal, Mitigating address spoofing attacks in hybrid SDN, Int. J. Adv. Comput. Sci. Appl. 8 (4) (2017) 562–570, http://dx.doi.org/10.14569/IJACSA.2017.080474.

[79] T. Alharbi, D. Durando, F. Pakzad, M. Portmann, Securing ARP in software defined networks, in: IEEE 41st Conference on Local Computer Networks, LCN 2016, 2016, pp. 523–526, http://dx.doi.org/10.1109/LCN.2016.83.

[80] Y. Lu, M. Wang, P. Huang, An SDN-based authentication mechanism for securing neighbor discovery protocol in ipv6, Secur. Commun. Netw. 2017 (2017) 1–9, http://dx.doi.org/10.1155/2017/5838657.

[81] D.M.F. Mattos, O.C.M.B. Duarte, AuthFlow: authentication and access control mechanism for software defined networking, Ann. Telecommun. 71 (11–12) (2016) 607–615, http://dx.doi.org/10.1007/s12243-016-0505-z.

[82] F. Kuliesius, V. Dangovas, SDN enhanced campus network authentication and access control system, in: Eighth International Conference on Ubiquitous and Future Networks, ICUFN 2016, 2016, pp. 894–899, http://dx.doi.org/10.1109/ICUFN.2016.7536925.

[83] J. Kwon, D. Seo, M. Kwon, H. Lee, A. Perrig, H. Kim, An incrementally deployable anti-spoofing mechanism for software-defined networks, Comput. Commun. 64 (2015) 1–20, http://dx.doi.org/10.1016/j.comcom.2015.03.003.

[84] B. Liu, J. Bi, Y. Zhou, Source address validation in software defined networks, in: ACM SIGCOMM Conference, 2016, pp. 595–596, http://dx.doi.org/10.1145/2934872.2960425.

[85] G. Yao, J. Bi, P. Xiao, Source address validation solution with openflow/NOX architecture, in: International Conference on Network Protocols, ICNP 2011, 2011, pp. 7–12, http://dx.doi.org/10.1109/ICNP.2011.6089085.

[86] G. Yao, J. Bi, T. Feng, P. Xiao, D. Zhou, Performing software defined route-based IP spoofing filtering with SEFA, in: International Conference on Computer Communications and Networks, ICCCN 2014, 2014, pp. 1–8, http://dx.doi.org/10.1109/ICCCN.2014.6911784.

[87] J.H. Jafarian, E. Al-Shaer, Q. Duan, Formal approach for route agility against persistent attackers, in: Lecture Notes in Computer Science, Vol. 8134, Springer International Publishing, 2013, pp. 237–254, http://dx.doi.org/10.1007/978-3-642-40203-6_14.

[88] F. Gillani, E. Al-Shaer, S. Lo, Q. Duan, M. Ammar, E. Zegura, Agile virtualized infrastructure to proactively defend against cyber attacks, in: IEEE Conference on Computer Communications, INFOCOM 2015, 2015, pp. 729–737, http://dx.doi.org/10.1109/INFOCOM.2015.7218442.

[89] S. Fichera, L. Galluccio, S.C. Grancagnolo, G. Morabito, S. Palazzo, OPERETTA: An openflow-based remedy to mitigate TCP SYNFLOOD attacks against web servers, Comput. Netw. 92 (2015) 89–100, http://dx.doi.org/10.1016/j.comnet.2015.08.038.

[90] S. Shin, V. Yegneswaran, P. Porras, G. Gu, AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks, in: ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, 2013, pp. 413–424, http://dx.doi.org/10.1145/2508859.2516684.

[91] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Comput. Netw. 62 (2014) 122–136, http://dx.doi.org/10.1016/j.bjp.2013.10.014.

[92] A. Hussein, I.H. Elhajj, A. Chehab, A. Kayssi, SDN security plane: An architecture for resilient security services, in: IEEE International Conference on Cloud Engineering Workshop, IC2EW 2016, 2016, pp. 54–59, http://dx.doi.org/10.1109/IC2EW.2016.15.

[93] O. Joldzic, Z. Djuric, P. Vuletic, A transparent and scalable anomaly-based dos detection method, Comput. Netw. 104 (2016) 27–42, http://dx.doi.org/10.1016/j.comnet.2016.05.004.

[94] J. Li, S. Berg, M. Zhang, P. Reiher, T. Wei, Drawbridge-software-defined DDoS-resistant traffic engineering, ACM SIGCOMM Comput. Commun. Rev. 44 (4) (2014) 591–592, http://dx.doi.org/10.1145/2619239.2631469.

[95] R. Miao, M. Yu, N. Jain, NIMBUS: Cloud-scale attack detection and mitigation, ACM SIGCOMM Comput. Commun. Rev. 44 (2014) 121–122, http://dx.doi.org/10.1145/2740070.2631446.

[96] Y.E. Oktian, S. Lee, H. Lee, Mitigating denial of service (DoS) attacks in openflow networks, in: International Conference on Information and Communication Technology Convergence, ICTC 2014, 2014, pp. 325–330, http://dx.doi.org/10.1109/ICTC.2014.6983147.

[97] A.F.M. Piedrahita, S. Rueda, D.M.F. Mattos, O.C.M.B. Duarte, Flowfence: A denial of service defense system for software defined networking, in: Global Information Infrastructure and Networking Symposium, GIIS 2015, 2015, pp. 1–6, http://dx.doi.org/10.1109/GIIS.2015.7347185.

[98] T. Wang, H. Chen, G. Cheng, Y. Lu, SDNManager: A safeguard architecture for SDN DoS attacks based on bandwidth prediction, Secur. Commun. Netw. 2018 (2018) http://dx.doi.org/10.1155/2018/7545079.

[99] B. Wang, Y. Zheng, W. Lou, Y.T. Hou, DDoS attack protection in the era of cloud computing and software-defined networking, Comput. Netw. 81 (2015) 308–319, http://dx.doi.org/10.1016/j.comnet.2015.02.026.

[100] G. Shang, P. Zhe, X. Bin, H. Aiqun, R. Kui, Flooddefender: Protecting data and control plane resources under SDN-aimed DoS attacks, in: INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, IEEE, 2017, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2017.8057009.

[101] S. Wang, S. Chandrasekharan, K. Gomez, S. Kandeepan, A. Al-Hourani, M.R. Asghar, G. Russello, P. Zanna, SECOD: SDN secure control and data plane algorithm for detecting and defending against DoS attacks, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–5, http://dx.doi.org/10.1109/NOMS.2018.8406196.

[102] N. Goksel, M. Demirci, Dos attack detection using packet statistics in SDN, in: International Symposium on Networks, Computers and Communications, ISNCC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ISNCC.2019.8909114.

[103] R. Xie, M. Xu, J. Cao, Q. Li, Softguard: Defend against the low-rate TCP attack in SDN, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761806.

[104] L. Wang, Q. Li, Y. Jiang, J. Wu, Towards mitigating link flooding attack via incremental SDN deployment, in: IEEE Symposium on Computers and Communication, ISCC 2016, 2016, pp. 397–402, http://dx.doi.org/10.1109/ISCC.2016.7543772.

[105] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, F. Yu, Detecting and mitigating target link-flooding attacks using SDN, IEEE Trans. Dependable Secure Comput. 16 (6) (2019) 944–956, http://dx.doi.org/10.1109/TDSC.2018.2822275.

[106] J. Zheng, Q. Li, G. Gu, J. Cao, D.K. Yau, J. Wu, Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis, IEEE Trans. Inf. Forensics Secur. 13 (7) (2018) 1838–1853.

[107] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, L. Gong, Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN, Int. J. Commun. Syst. 31 (5) (2018) e3497, http://dx.doi.org/10.1002/dac.3497.

[108] J. Cui, J. He, Y. Xu, H. Zhong, TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller, in: Australasian Conference on Information Security and Privacy, Springer, 2018, pp. 649–665, http://dx.doi.org/10.1007/978-3-319-93638-3_37.

[109] Z.A. El Houda, A.S. Hafid, L. Khoukhi, Cochain-SC: An intra-and inter-domain ddos mitigation scheme based on blockchain using SDN and smart contract, IEEE Access 7 (2019) 98893–98907, http://dx.doi.org/10.1109/ACCESS.2019.2930715.

[110] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, B. Stiller, A blockchain-based architecture for collaborative DDoS mitigation with smart contracts, in: IFIP International Conference on Autonomous Infrastructure, Management and Security, Springer, Cham, 2017, pp. 16–29, http://dx.doi.org/10.1007/978-3-319-60774-0_2.

[111] Z. Abou El Houda, A. Hafid, L. Khoukhi, Co-IoT: A collaborative ddos mitigation scheme in IoT environment based on blockchain using SDN, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013542.

[112] J. Xing, W. Wu, A. Chen, Architecting programmable data plane defenses into the network with fastflex, in: Proceedings of the 18th ACM Workshop on Hot Topics in Networks, 2019, pp. 161–169.

[113] M. Zhang, G. Li, S. Wang, C. Liu, A. Chen, H. Hu, G. Gu, Q. Li, M. Xu, J. Wu, Poseidon: Mitigating volumetric DDoS attacks with programmable switches, in: Proceedings of NDSS, 2020.

[114] SFlow, 2020, http://sflow.org, (Accessed: 22 May 2020).

[115] M.S. Kang, S.B. Lee, V.D. Gligor, The crossfire attack, in: Security and Privacy (SP), 2013 IEEE Symposium on, IEEE, 2013, pp. 127–141, http://dx.doi.org/10.1109/SP.2013.19.

[116] Project floodlight, 2020, https://github.com/floodlight/, (Accessed: 22 May 2020).

[117] R. Singh, S. Tanwar, T.P. Sharma, Utilization of blockchain for mitigating the distributed denial of service attacks, Secur. Priv. 3 (3) (2020) e96, http://dx.doi.org/10.1002/spy2.96.

[118] Z. Zhao, D. Gong, B. Lu, F. Liu, C. Zhang, SDN-based double hopping communication against sniffer attack, Math. Probl. Eng. 2016 (2016) 1–13, http://dx.doi.org/10.1155/2016/8927169.

[119] Q. Duan, E. Al-Shaer, H. Jafarian, Efficient random route mutation considering flow and network constraints, in: IEEE Conference on Communications and Network Security, CNS 2013, 2013, pp. 260–268, http://dx.doi.org/10.1109/CNS.2013.6682715.

[120] J. Liu, H. Zhang, Z. Guo, A defense mechanism of random routing mutation in SDN, IEICE Trans. Inf. Syst. E100D (5) (2017) 1046–1054, http://dx.doi.org/10.1587/transinf.2016EDP7377.

[121] M. Furukawa, K. Kuroda, T. Ogawa, N. Miyaho, Highly secure communication service architecture using SDN switch, in: 10th Asia-Pacific Symposium on Information and Telecommunication Technologies, APSITT 2015, 2015, pp. 1–3, http://dx.doi.org/10.1109/APSITT.2015.7217098.

[122] D. Ma, L. Wang, C. Lei, Z. Xu, H. Zhang, M. Li, Thwart eavesdropping attacks on network communication based on moving target defense, in: IEEE 35th International Performance Computing and Communications Conference, PCCC 2016, 2016, pp. 1–2, http://dx.doi.org/10.1109/PCCC.2016.7820610.

[123] E. Germano Da Silva, L.A. Dias Knob, J.A. Wickboldt, L.P. Gaspary, L.Z. Granville, A. Schaeffer-Filho, Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study, in: IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, 2015, pp. 165–173, http://dx.doi.org/10.1109/INM.2015.7140289.

[124] B. Villain, J. Ridoux, J. Rotrou, G. Pujolle, Mutualized openflow architecture for network access management, in: IEEE 3rd International Conference on Cloud Networking, CloudNet 2014, 2014, pp. 413–419, http://dx.doi.org/10.1109/CloudNet.2014.6969030.

[125] K. Cabaj, W. Mazurczyk, Using software-defined networking for ransomware mitigation: The case of cryptowall, IEEE Netw. 30 (6) (2016) 14–20, http://dx.doi.org/10.1109/MNET.2016.1600110NM.

[126] J.M. Ceron, B.M. Margi, L.Z. Granville, L. Gualberto, B. Goncalves, MARS: An SDN-based malware analysis solution, in: IEEE Symposium on Computers and Communication, ISCC 2016, 2016, pp. 525–530, http://dx.doi.org/10.1109/ISCC.2016.7543792.

[127] Y. Hu, K. Zheng, X. Wang, Y. Yang, WORM-HUNTER: A worm guard system using software-defined networking, KSII Trans. Internet Inf. Syst. 11 (1) (2017) 484–510, http://dx.doi.org/10.3837/tiis.2017.01.026.

[128] R. Jin, B. Wang, Malware detection for mobile devices using software-defined networking, in: Second GENI Research and Educational Experiment Workshop, GREE2013, 2013, pp. 81–88, http://dx.doi.org/10.1109/GREE.2013.24.

[129] M. Masoud, Y. Jaradat, A.Q. Ahmad, On tackling social engineering web phishing attacks utilizing software defined networks (SDN) approach, in: 2nd International Conference on Open Source Software Computing, OSSCOM 2016, 2016, pp. 1–6, http://dx.doi.org/10.1109/OSSCOM.2016.7863679.

[130] T. Chin, K. Xiong, C. Hu, Phishlimiter: a phishing detection and mitigation approach using software-defined networking, IEEE Access 6 (2018) 42516–42531, http://dx.doi.org/10.1109/ACCESS.2018.2837889.

[131] S. Lim, J. Ha, H. Kim, Y. Kim, S. Yang, A SDN-oriented DDoS blocking scheme for botnet-based attacks, in: 6th International Conference on Ubiquitous and Future Networks, ICUFN 2014, 2014, pp. 63–68, http://dx.doi.org/10.1109/ICUFN.2014.6876752.

[132] M. Shtern, R. Sandel, M. Litoiu, C. Bachalo, V. Theodorou, Towards mitigation of low and slow application ddos attacks, in: IEEE International Conference on Cloud Engineering, IC2E 2014, 2014, pp. 604–609, http://dx.doi.org/10.1109/IC2E.2014.38.

[133] O. Yurekten, M. Demirci, Using cyber threat intelligence in SDN security, in: 2017 International Conference on Computer Science and Engineering, UBMK, IEEE, 2017, pp. 377–382, http://dx.doi.org/10.1109/UBMK.2017.8093415.

[134] T. Bakhshi, State of the art and recent research advances in software defined networking, Wirel. Commun. Mob. Comput. 2017 (2017) http://dx.doi.org/10.1155/2017/7191647.

[135] The POX network software platform, 2020, https://github.com/noxrepo/pox/, (Accessed: 22 May 2020).

[136] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, NOX: towards an operating system for networks, ACM SIGCOMM Comput. Commun. Rev. 38 (3) (2008) 105–110, http://dx.doi.org/10.1145/1384609.1384625.

[137] Ryu SDN framework, 2020, https://osrg.github.io/ryu/, (Accessed: 22 May 2020).

[138] Mininet: An instant virtual network on your laptop (or other PC), 2020, http://mininet.org/, (Accessed: 22 May 2020).

[139] Open vswitch, 2020, https://www.openvswitch.org/, (Accessed: 22 May 2020).

[140] H. Hu, Z. Wang, G. Cheng, J. Wu, MNOS: a mimic network operating system for software defined networks, IET Inf. Secur. 11 (6) (2017) 345–355, http://dx.doi.org/10.1049/iet-ifs.2017.0085.

[141] P.K. Sharma, S. Singh, Y.-S. Jeong, J.H. Park, Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks, IEEE Commun. Mag. 55 (9) (2017) 78–85, http://dx.doi.org/10.1109/MCOM.2017.1700041.

[142] H. Yang, J. Yuan, H. Yao, Q. Yao, A. Yu, J. Zhang, Blockchain-based hierarchical trust networking for jointcloud, IEEE Internet Things J. 7 (3) (2019) 1667–1677, http://dx.doi.org/10.1109/JIOT.2019.2961187.

[143] P.K. Sharma, J.H. Park, Blockchain based hybrid network architecture for the smart city, Future Gener. Comput. Syst. 86 (2018) 650–655, http://dx.doi.org/10.1016/j.future.2018.04.060.

[144] H. Yang, Y. Liang, J. Yuan, Q. Yao, A. Yu, J. Zhang, Distributed blockchain-based trusted multi-domain collaboration for mobile edge computing in 5G and beyond, IEEE Trans. Ind. Inf. (2020) http://dx.doi.org/10.1109/TII.2020.2964563.

[145] L. Xie, Y. Ding, H. Yang, X. Wang, Blockchain-based secure and trustworthy internet of things in SDN-enabled 5G-VANETs, IEEE Access 7 (2019) 56656–56666, http://dx.doi.org/10.1109/ACCESS.2019.2913682.

[146] H. Yang, Y. Liang, Q. Yao, S. Guo, A. Yu, J. Zhang, Blockchain-based secure distributed control for software defined optical networking, China Commun. 16 (6) (2019) 42–54, http://dx.doi.org/10.23919/JCC.2019.06.004.

[147] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the Internet of Things: A comprehensive survey, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1676–1717, http://dx.doi.org/10.1109/COMST.2018.2886932.

[148] F.H. Pohrmen, R.K. Das, G. Saha, Blockchain-based security aspects in heterogeneous Internet-of-Things networks: A survey, Trans. Emerg. Telecommun. Technol. 30 (10) (2019) e3741, http://dx.doi.org/10.1002/ett.3741.

[149] G. Praveen, V. Chamola, V. Hassija, N. Kumar, Blockchain for 5G: A prelude to future telecommunication, IEEE Network (2020) http://dx.doi.org/10.1109/MNET.001.2000005.

[150] W. Connell, D.A. Menasce, M. Albanese, Performance modeling of moving target defenses with reconfiguration limits, IEEE Trans. Dependable Secure Comput. (2018) http://dx.doi.org/10.1109/TDSC.2018.2882825.

Ozgur Yurekten is a chief researcher with Cyber Security Institute, TUBITAK-BILGEM in Turkey. He has worked on software development, digital transformation, and cyber security projects. He is pursuing his Ph.D. in Computer Engineering at Gazi University in Ankara, Turkey. He received his B.S. degree in Computer Engineering from Baskent University in Ankara, Turkey in 2001, and his M.S. degree in Computer Engineering from Gazi University in 2007. His research interests include cyber security, software defined networking, cloud computing, software engineering, secure software development, software architectures, and digital transformation.

Mehmet Demirci received his B.S. degree in computer science and mathematics (double major) from Purdue University, West Lafayette, Indiana, USA in 2006, and his M.Sc. and Ph.D. degrees in computer science from Georgia Institute of Technology, Atlanta, Georgia, USA, in 2009 and 2013, respectively. He is currently an assistant professor with the Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey. His current research interests are software-defined networking (SDN), network functions virtualization (NFV), 5G, network security, network architecture & future Internet. Dr. Demirci has authored research articles in international journals and conference proceedings, such as IEEE Transactions on Network and Service Management (TNSM), Computer Communications, IEEE GLOBECOM, IEEE MASCOTS, CNSM, and IEEE ICMLA. He serves as the NFV-SDN track chair in the International Symposium on Networks, Computers, and Communications (ISNCC) sponsored by IEEE.