IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 May 2024, accepted 10 July 2024, date of publication 16 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3429269

SURVEY

On the Road to Proactive Vulnerability Analysis and Mitigation Leveraged by Software Defined Networks: A Systematic Review

JOÃO POLÓNIO[1], **JOSÉ MOURA**[1,2], **AND RUI NETO MARINHEIRO**[1,2]
[1]Instituto Universitário de Lisboa (ISCTE–IUL), 1649-026 Lisbon, Portugal
[2]Instituto de Telecomunições (IT), 1049-001 Lisbon, Portugal

Corresponding author: José Moura (jose.moura@iscte-iul.pt)

This work was supported in part by Fundação para a Ciência e Tecnologia/Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES) through National Funds When Applicable Co-Funded European Union (EU) Funds under Project UIDB/50008/2020; and in part by the Instituto de Telecomunicações, Lisbon, Portugal.

ABSTRACT The discovery of security vulnerabilities and their mitigation in networked systems managed by Software-Defined Networking (SDN) are fundamental for ensuring their normal operation. The main goal of this survey was to investigate the literature on preventing system security vulnerabilities instead of detecting ongoing cyber-attacks as quickly as possible. Thus, in our opinion, organizations should fortify their systems' security by identifying and eliminating any new security vulnerabilities before they can be successfully exploited. We comprehensively discuss different vulnerability detection approaches based on important comparison parameters such as vulnerability assessment, the SDN controller used, automation capability, system risk indicators, passive scanning and active probing of system vulnerabilities. The paper also analyzes relevant literature considering the mitigation mechanisms for discovered vulnerabilities such as the proposed SDN controller, automation capability, solution adaptation to system operational changes, risk indicators, and the solution's impact on network quality metrics like latency and throughput. Despite the strengths of the surveyed work, we have also identified promising open issues that need further consideration by scholars, industry participants, and policymakers. We concluded that the majority of analyzed literature contributions are largely reactive in their implementation against running network threats. This suggests a new research domain for applying SDN in the automatic detection of security vulnerabilities and their proactive mitigation before external cyber-attackers can exploit them.

INDEX TERMS System vulnerability, assessment, detection, mitigation, software defined networks, risk evaluation, automatic operation, network security.

I. INTRODUCTION

Networks are growing every day, and the security of networked systems is becoming an increasingly important need to account for. Hackers are well-known for posing a serious security risk, as on a daily basis, security firms and mass media sources report a rising number of sophisticated cyber attacks. Projections indicate that the worldwide expense associated with cyber crime will soar over the next years, escalating from $8.44 trillion in 2022 to $23.82 trillion by 2027, as represented in Fig. 1 [1].

Normally an attacker has an enduring economic advantage over the system defender, because the defender needs to fix all the system vulnerabilities, while the attacker only needs to exploit one of those vulnerabilities with success. Considering this economic asymmetry, the greater the difficulty for the human defender to keep the system safe and the ever-evolving landscape of cyber security threats against networked systems, it is imperative to delve deeper into the research on automatic and proactive solutions for

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Giorgetti.

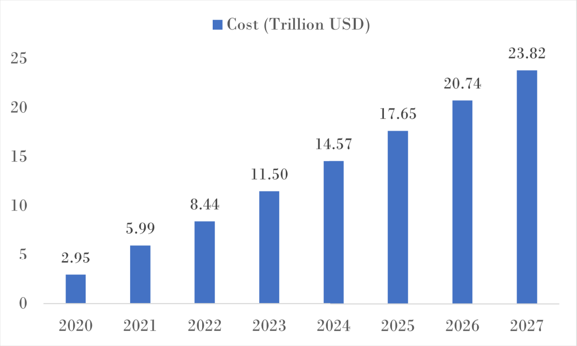

FIGURE 1. Estimated cost of cyber crime worldwide [1].

discovering and mitigating system security vulnerabilities, diminishing the risk of system normal operation being jeopardized. In this way, we have decided to drive a comprehensive and systematic literature revision on these subjects. The papers of the current survey were collected using the Systematic Literature Review (SLR), which involves the Planning, Conducting, and Reporting phases [2]. To this end, Parsifal [3] was used to identify and evaluate all the most relevant literature on each paper topic. The Parsifal tool has been used in several related literature, such as [4], [5], and [6].

During our investigation, the papers initially retrieved were based on the search string shown in Fig. 2. Then, the initially found literature was further filtered and aligned, considering the current survey's main goals, which are reflected in the selection criteria of Table 1. After this second phase, Table 2 shows the number of publications returned by the search string (Fig. 2) and the final papers accepted for critical analysis during our manuscript. The average paper acceptance ratio was around 5%. In addition, Fig. 3 shows the trend on the number of publications retrieved by the search string between 2016 and 2023.

In the search string of Fig. 2, the first term regarding SDN aimed to mainly discover literature solutions leveraged by the SDN paradigm. The term after AND encompasses the two domains covered in this survey, namely Vulnerability Detection and Mitigation. Considering the case of detection, the search was restrained to works that also mentioned vulnerabilities, while in the case of mitigation, a more generic search was opted for, since some attack detection contributions were relevant to be included in this survey. The search string was enhanced by adding a set of techniques/technologies, which were separated by ORs, as the goal was to find papers discriminated by the used technique to protect the system operation. Thus, our search string aimed to find out a well representative sample of the more recent literature in the survey investigation area, and then, carry out a comprehensive analysis, discussion and interesting comparison among the found work to highlight research open issues. Survey [7] shows similar concerns regarding the use of some terms in its search string regarding SDN, security, vulnerability and mitigation.

TABLE 1. Selection criteria.

Inclusion Criteria	Exclusion Criteria
Pioneer Number of Citations	Prior to 2016 Out of Scope SDN Architecture Attack-based Investigation Studies not written in English

("SDN" OR "Software Defined Networks" OR "Software Defined Networking") AND ("Vulnerability Scanner" OR "Vulnerabilities Scanning" OR "Vulnerabilities Scanner" OR "Vulnerabilities Detection" OR "Vulnerability Detection" OR "Mitigation" OR "MTD" OR "Moving Target Defense" OR "AG" OR "Attack Graph" OR "Attack Graphs" OR "Honeypot" OR "Honeynet" OR ("P4" AND "Security"))

FIGURE 2. Literature review search string.

TABLE 2. Accepted related work.

Digital Library	Imported Studies	Accepted	Percentage
IEEEXplore	763	46	6.02%
ACM Digital Library	216	6	2.78%

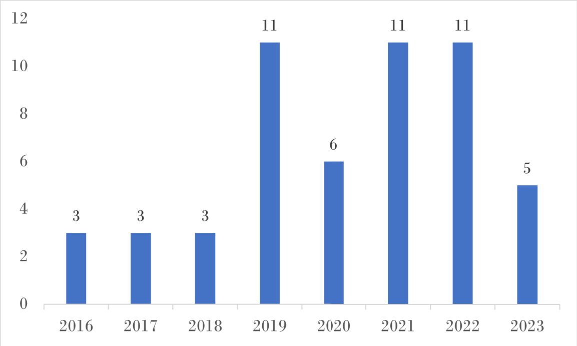

FIGURE 3. Post-2016 analyzed detection and mitigation papers per Year.

A very interesting outcome from our literature analysis is that there is a strong need for further investigation on detecting and mitigating vulnerabilities at networked systems [8], [9]. Although there is a lot of research in the area of SDN to detect and mitigate different types of attacks, there is not the same degree of concern about the attack prevention phase. Considering our current best knowledge, this survey is the first tentative to frame proactive solutions for detecting and mitigating vulnerabilities in network systems controlled by SDN. We have excluded from our analysis the secure delivery of online content via either IP [10] or Information Centric [11] networks as well as the SDN security [12], [13]. The paper contributions are as follows:

- Analysis and comparison of work concerned with system vulnerabilities detection and their risk against the system normal operation;

- Proximity score with the current survey scope of available related literature and clear highlight of the most prominent proposals;
- Discuss the integration of proactive and automatic attack deception technologies for minimizing the security risk of future threats exploring system vulnerabilities;
- Identify some guidelines for future research developments in the areas covered by this work.

The rest of the paper is organized as Fig. 4. Section II briefly contextualizes the work, presents SDN as a comprehensive approach to enhance the operation of networking infrastructures as well as the most relevant technologies and security vulnerability assessment metrics that will be discussed along the survey. Section III addresses research on detecting system vulnerabilities, which could be potentially explored by threats against the normal system operation. Section IV compares and analyzes several proactive actions to minimize the security risk of future cyber attacks exploring system vulnerabilities. Section V discusses relevant open issues that may serve as guidelines for further research on the main survey topics. Section VI concludes the manuscript. The Table 3 lists the acronyms used in the paper.

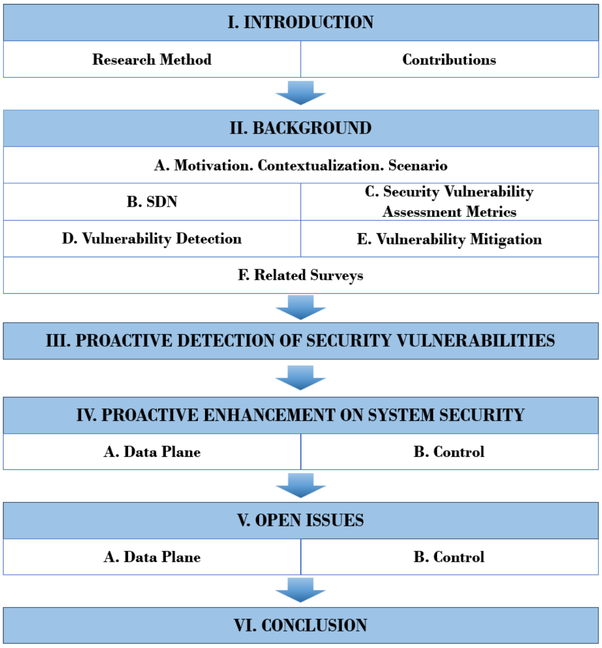
I. INTRODUCTION
Research Method
Contributions
II. BACKGROUND
A. Motivation. Contextualization. Scenario
B. SDN
C. Security Vulnerability Assessment Metrics
D. Vulnerability Detection
E. Vulnerability Mitigation
F. Related Surveys
III. PROACTIVE DETECTION OF SECURITY VULNERABILITIES
IV. PROACTIVE ENHANCEMENT ON SYSTEM SECURITY
A. Data Plane
B. Control
V. OPEN ISSUES
A. Data Plane
B. Control
VI. CONCLUSION

FIGURE 4. Article structure.

TABLE 3. Paper acronyms.

ACL	Access Control List
AG	Attack Graph
ASP	Attack Success Probability
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
DPI	Deep Packet Inspection
DRL	Deep Reinforcement Learning
DRRM	Dynamic Random Route Mutation
FL	Floodlight
FRVM	Flexible Random Virtual IP Multiplexing
FW	Firewall
GVM	Greenbone Vulnerability Management
GSPN	Generalized Stochastic Petri Net
HP	Honeypot
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
MADS	MTD Adaptive Delay System
ML	Machine Learning
MTD	Moving Target Defense
NBI	Northbound Interface
ODL	OpenDayLight
ONOS	Open Network Operating System
PaH	Path Hopping
PoH	Port Hopping
PHCSS	Port Hopping technique with Masked Communication Services
PH-SND	Path Hopping Based SDN Network Defense Technology
PK	Port Knocking
SBI	Southbound Interface
SCEMA	SDN-oriented Cost-effective Edge-based MTD Approach
SDN	Software-Defined Networking
SOAR	Security Orchestration, Automation, and Response
TCP	Transmission Control Protocol
TV-HARM	Threat Vector Hierarchical Attack Representation Mode
VLAN	Virtual Local Area Network
ZAP	Zed Attack Proxy

II. BACKGROUND

The previous section introduced the topic covered in this article and its main contributions, as well as the used literature review method. This section discusses the Motivation, Context and Scenario for this survey, providing the necessary framework for understanding the practical implications and drivers of the literature. Next, we will briefly explain the main concepts covered in the literature, which are SDN, security vulnerability assessment metrics, and techniques for vulnerability detection and mitigation. The concept of SDN is explained to promote a better understanding of the solutions presented in the literature. The security vulnerability assessment metrics we discuss are used as a standard to identify and assess the vulnerabilities found in systems. In vulnerability detection we explain the different types of vulnerability detection and how as a more complete analysis of the vulnerabilities found can be useful. Vulnerability mitigation defines the different techniques that have been employed in the literature to react to vulnerabilities that are detected. We also compare our survey with others to identify holes or areas for research and ensure the validity and relevance of our contributions to the literature.

A. MOTIVATION. CONTEXTUALIZATION. SCENARIO

Computer networks have become increasingly complex due to the mass dissemination of content and services, the introduction of sophisticated applications, and the growth of the Internet of Things (IoT), which is leading to an exponential increase on the number of devices at the networked systems. Controlling a such substantial number of devices is difficult and prone to errors. In addition, the traditional networks are hardware-centric and have serious problems with research and innovation, flexibility, and manageability. So, the need for networked systems with greater capacity, better accessibility, and dynamic resource management is turning into a crucial issue [14].

Two very important points should be mentioned to justify why SDN may be a beneficial option for offering innovative solutions to some of the traditional network issues. The first point is that SDN integrates many technologies into a unified network, resulting in a more flexible and scalable solution for heterogeneous networks. The second point is that the SDN centralized logical control plane enables faster network reconfiguration without impacting the underlying controlled network devices.

The topic of this survey falls within the context of the proactive vulnerability assessment and mitigation of threats where SDN capabilities will be leveraged. To illustrate the importance of proactive management solutions, an example would be an institution concerned about keeping its networked system as secure as possible. In this scenario, the internal hosts operating systems have frequent updates, which can make these hosts vulnerable to external threats, and thus the institution system suddenly become more vulnerable to attacks. The value of integrating SDN with security systems is transpired by the SDN ability to program the network and automatically support security-related tasks such as vulnerabilities assessment, including their risk, and incident response. The visibility SDN controllers have from the network operation may turn possible to not only monitor suspect network traffic in real-time but also automatically isolate some discovered compromised network devices inside a specific VLAN with limited access to important system operational parts. Therefore, the SDN is a flexible and scalable solution for heterogeneous networks, enabling the network reconfiguration without impacting the underlying devices, and protecting the normal network operation.

B. SOFTWARE-DEFINED NETWORKING

The physical separation between the control plane and the forwarding plane is the key feature of the SDN architecture. The network's state is maintained by a logically centralized control function, which also gives instructions to the data plane. This separation is essential to achieving the necessary flexibility because it breaks the network control problem into manageable chunks, turns simpler to develop and implement new networking abstractions, and promotes network growth and innovation [15]. Fig. 5 shows the seven main components that make up the SDN architecture, namely the three planes Data, Control, and Management, as well as the Northbound, Southbound, and East/Westbound interfaces.

The management plane is the topmost plane of the SDN architecture and aggregates several applications with very distinct responsibilities such as routing, load balancing or security. The management plane is a crucial component of an SDN design because it optimizes, with the maximum abstraction from the network complexity, the global system operation using specialized software and tools. Through the northbound interface (NBI) of the SDN architecture, the management plane communicates with the control plane, enabling applications to submit commands and receive data from the control plane [14].

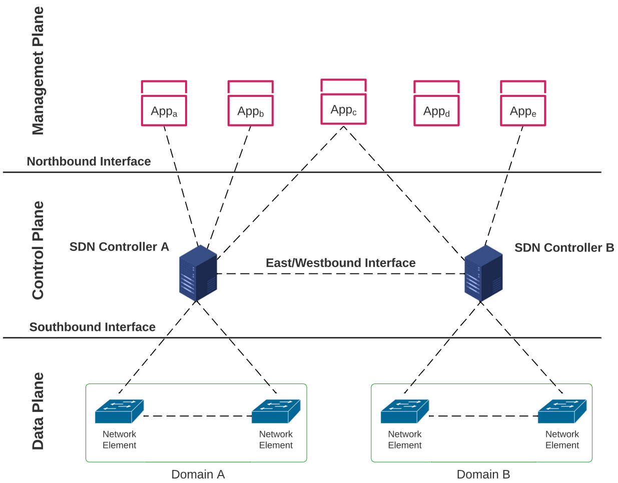

FIGURE 5. SDN architecture.

The control plane oversees and regulates the network's data flow. Through a specified southbound interface (SBI) control plane components can program the behavior of data plane forwarding devices. Depending on several factors, including the packet's destination, traffic type, and the network's resources, every SDN controller is in charge of deciding how message data should be sent via the data plane. The information about the state of the network retrieved via SBI is used by the control plane to decide on routing and set up network devices to forward traffic efficiently through the networking infrastructure. The SDN controller enables the network to be more adaptable and agile by allowing the control of the network to be separated from the data plane devices. This facilitates the use of specialized software and tools to optimize the network and makes it simpler to administer and reconfigure the network as necessary. To enhance the performance and reliability of the logical centralized control level, several SDN controllers should decide in parallel about how the network resources should be used. Thus, some coordination among the SDN controllers is required for networks controlled by multiple SDN controllers. The more correct orchestration among the several SDN controllers can be supported by East-West interfaces [14].

The data plane is the bottom-most plane of the SDN architecture and is formed by the forwarding devices as switches and routers. These network elements forward the data plane messages according to the flow rules, which are dynamically installed and updated at those network elements via OpenFlow protocol by the upper-plane SDN controllers. Alternatively, in P4 programmable data plane devices, the forwarding behavior is specified in devices scripts [16]. After the compilation of each script, it is generated a low-level firmware code compatible with the network device where that code will be running as well as an intermediate level code for the device (i.e. client) OpenFlow API to ensure a proper comunication with the upper level SDN controller. This allows to modify and adapt the behavior of the data plane to meet new operational capabilities [17], which is a main advantage of using P4 in detriment of static and pre-defined behavior of pure OpenFlow devices.

C. SECURITY VULNERABILITY ASSESSMENT METRICS

This subsection explores the important topic of Security Vulnerability Assessment Metrics, which includes the Common Vulnerabilities and Exposures (CVE) system and the Common Vulnerability Scoring System (CVSS). These metrics are fundamental components to actively identify, assess, and ranking security vulnerabilities in information systems, considering their risk against the system normal operation. The CVE system offers a defined nomenclature to uniquely identify security vulnerabilities, promoting a shared comprehension within the cybersecurity community. Simultaneously, the CVSS offers a numerical method to evaluate the seriousness of these vulnerabilities, assisting in prioritizing their proper response activities [18].

1) COMMON VULNERABILITIES AND EXPOSURE ID

The Common Vulnerabilities and Exposure ID (CVE ID) provide a reliable method of identifying unique vulnerabilities and coordinating the development of security tools and solutions. CVE IDs are formatted as CVE-YYYY-NNNNN, where the YYYY part represents the year that the CVE ID was assigned or the vulnerability was made public. Security flaws that become CVE entries are frequently contributed by members of the open source community [19].

2) COMMON VULNERABILITY SCORING SYSTEM

The Common Vulnerability Scoring System (CVSS) aims to assign vulnerability severity scores, allowing to prioritize responses and resources based on the found threat. A CVSS score is made up of three sets of metrics (Base, Temporal, and Environmental), each with its own scoring component [20].

The Base metric group represents a vulnerability's intrinsic characteristic that remains consistent over time and across user environments. It is made up of two types of metrics: Exploitability metrics and Impact metrics [18]. The Exploitability metrics measure how easy and the necessary effort to exploit the vulnerability. The Impact metrics indicate the negative consequence on the target vulnerable system component after the vulnerability has been explored.

The Temporal metric group reflects vulnerability characteristics that may change over time but not across user environments. The inclusion of an exploit kit, for example, would raise the CVSS score, but the creation of an official patch would lower it [18].

The Environmental metric group represents susceptibility factors that are relevant and unique to a specific user's environment. The availability of security mechanisms that may minimize either partially or totally the negative consequences of a successful attack, as well as the relevance of a vulnerable component being present in a technological infrastructure, are all considered by the environmental metric [18].

Table 4 shows the Severity Rating Scale for CVSS v3.0 and v3.1 [18]. The system CVSS vulnerability severity is evaluated on a scale from 0.0 to 10.0, where a higher score indicates a more critical system vulnerability, meaning a vulnerability when explored by an attacker could produce a higher damage to the networked system. Thus, CVSS aids in prioritization among several system discovered vulnerabilities and next mitigation tasks.

TABLE 4. CVSS v3.0 and v3.1 severity rating scale.

Severity	None	Low	Medium	High	Critical
Base Score	0.0	0.1-3.9	4.0-6.9	7.0-8.9	9.0-10.0

D. VULNERABILITY DETECTION

This subsection discusses the main technologies used for vulnerability detection, whether active or passive. Attack Graphs (AGs) are considered by us to be a detection technology because they are often used by authors as a way of improving threat risk classification, but it is worth mentioning that this technology is also used to improve the effectiveness of mitigation technologies, which will be discussed later.

Passive Scanning consists of analyzing traffic passing through a network monitoring point. This monitorization is normally unnoticed by the hosts running the services [21]. Some examples of passive tools are Wireshark, Snort, TCPDUMP and Zeek. Active Probing involves interacting with services by sending packets to each host and monitoring their response [21]. Some examples of active tools are Nmap, Metasploit, hping, Zed Attack Proxy (ZAP), Burp Suite, Greenbone Vulnerability Management (GVM) and NESSUS. Attack Graphs can be an addition to these technologies since represent the relationship between different system security vulnerabilities that may be exploited by an attacker, as well as the corresponding system access privileges acquired by the attacker after the exploitation of system vulnerabilities. Various AGs can be created based on the representations of nodes and edges. An AG consists on several node types such as state, host, privilege, or vulnerability. When an attacker successfully exploits a vulnerability, it frequently results in the escalation of privileges on the affected hosts, granting to the attacker the root access in those hosts [22]. This enables the attacker to prepare the next attack phases.

E. VULNERABILITY MITIGATION

The importance of vulnerability mitigation techniques lies in safeguarding against and minimizing the system negative impact of potential security threats. Therefore, these techniques are extremely important for network security research. Fig. 6 provides a classification of mitigation techniques

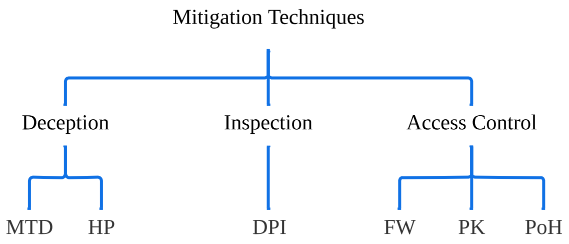

FIGURE 6. Classification of mitigation techniques.

further detailed and discussed in section IV. The deception was by far the most used mitigation type in the reviewed literature, but other interesting mitigation studies using Firewall, Port Knocking, Port Hopping and Deep Packet Inspection are also critically discussed in section IV.

The selection of the mitigation technique to be used on a particular use case controlled by SDN should be aware of that selected mitigation technique impact on the system performance and required system resources, as we following explain. For example, MTD is resource-intensive due to the frequent reconfiguration of network settings. However, applying MTD only to critical network subdomains may reduce the system overhead and keep its performance at a satisfactory level. Considering other alternative, such as DPI, it can also penalize the system performance due to real-time traffic analysis. Nevertheless, the negative impact on the system operation can be diminished by employing traffic sampling techniques, which analyze only a subset of traffic messages. Considering now PK, which typically consumes less resources than DPI, because the former controls the initial traffic access to a network domain via a correct sequence of used transport ports. In addition, the system performance may be protected by triggering PK operation via learned system events. The text below presents the found types of mitigation techniques from the analyzed literature.

1) MOVING TARGET DEFENSE

The goal of Moving Target Defense (MTD) is to randomly change the components of an underlying system. This guarantees that the information obtained by the attacker during the reconnaissance phase becomes stale during the attack phase since the defense has moved to a new configuration during that time. This creates confusion for the attackers, making it more difficult and expensive to properly exploit the system [23]. This dynamism requires the creation of a framework capable of accurately and timely examining the complex relationships between various hosts and security vulnerabilities, as well as ensuring that any changes made to the environment do not conflict with relevant active security policies neither penalize the system performance [24].

2) HONEYPOTS

Spitzner [25] gave one of the first formal definitions: “A honeypot is a decoy computer resource whose value lies in being probed, attacked, or compromised”. A Honeypot (HP) inhibits attacks because attackers waste time and money targeting honeypots rather than the real target. Honeypots can also accelerate the reaction to attacks, because attack traffic becomes isolated from the production traffic, turning the attack detection and analysis much more easy for the system defense. Additionally, HPs may be taken totally down for inspection, allowing for a complete forensic examination. The insights may then be utilized to clean up production systems and understand the exploit, which is the first relevant step towards patching the corresponding vulnerabilities [26]. Honeynets are entire decoy networks made out of one or more HPs.

3) DEEP PACKET INSPECTION

DPI enables network traffic analysis and eventual posterior traffic filtering. Thus, DPI identifies the data portion of a packet, which refers to its content, as well as DPI classifies the traffic via a signature, which corresponds to the packet’s ID. DPI devices analyze streaming packets to identify protocol non-compliant situations or domain intrusions. In the case of an incorrect traffic behavior, the associated packets should be dropped or deviated to an alternative destination for further inspection [27]. We see this technology as a very interesting way of mitigating vulnerabilities, but the available work only investigated it in a superficial way, requiring much more future work, as suggested in sub-section V-A.

4) FIREWALLS

Ensures the protection of the internal network’s security against external network attacks. The system employs established rules to selectively filter and regulate both incoming and outgoing traffic. The SDN switch may be configured to utilize the firewall (FW) functionality. To do this, one must incorporate the corresponding rules and action rules into the flow table of SDN switches [28].

5) PORT KNOCKING

Port Knocking (PK) is a technique used to externally unlock ports on a firewall. This is done by initiating a connection attempt on a predetermined list of locked ports. PK is an authentication mechanism utilized to transfer data via a closed port. Upon receiving a proper sequence of connection attempts, the firewall rules are adjusted in real-time to let the host that initiated the connection attempts to establish a connection through certain port(s). Once the secure authentication sequence is successfully executed, the server initiates the opening of a port exclusively for the authorized user, therefore establishing a secure and reliable connection between the client and the server [29]. Consequently, an attacker unaware about the correct port knocking sequence cannot directly monitor the server via reconnaissance methods.

6) PORT HOPPING

The fundamental concept behind port hopping (PoH) is the dynamic alteration of port numbers for important nodes.

Static ports can provide attackers with the opportunity to gradually acquire knowledge about the features of each service port during the reconnaissance attack phase. However, as the ports evolve over time, it becomes challenging to carry out an attack. The benefits of port hopping are straightforward and achievable, without the need for any modifications to existing protocols [30].

F. RELATED SURVEYS

We discuss now related surveys. The available literature is more concerned in discovering attacks [31], [32], [33] and less in preventing them. There is a lot of research that proposes improvements to the SDN architecture as shown in [7], [13], [34], [35], and [36], but only few works that proposes new ways of protecting the network by detecting and mitigating hosts vulnerabilities before they become major problems. Survey [37] is interesting because it deals in depth with automation that is one of the parameters used to compare the literature collected in our paper. In addition, [38] was the survey we have found more similar to ours, but with the difference the former be more focused in scenarios involving IoT and ours has a more wide network scope.

Survey [37] acknowledges the limitations of manual security operations and posits SDN as a solution that minimizes human error through its inherent design for minimal human intervention. The survey categorizes various security solutions based on their automation level and complexity. The automation is assessed using qualitative parameters like self-healing, self-adaptation, self-configuration, and self-optimization, while complexity is gauged by the resources and implementation requirements. A classification of the security solutions reveals the automation level enabled by each strategy and the complexity related to its implementation. The authors establish a collection of parameters and metrics to proficiently assess the network security design using SDN and shortly anticipate intelligent data planes to enhance the security of open, high-performance, and automated solutions.

Survey [38] offers an extensive examination of the security vulnerabilities present in IoT devices. The authors categorize the possible attack surfaces into three layers: Hardware, Software, and Protocol Interface. They emphasize that the attack surface extends due to the growing complexity and interactivity of the devices. This survey provides a detailed comparison and analysis of detection, discovery, and mitigation methods, categorizing them accordingly. The authors provide a comprehensive examination of vulnerability analysis technologies, focusing on four aspects: analysis tools, vulnerability discovery, detection, and mitigation. Furthermore, the study recognizes the difficulties presented by the heterogeneity of IoT devices, requiring the development of automated techniques for generating patches for multi-platform binary code. The survey emphasizes that future research should also focus on AI-based vulnerability discovery and detection, large-scale vulnerability analysis technique, among others.

The greatest contribution of our review in comparison to the existing literature is that the former comprehensively analyzes the literature, performing, as shown in Fig. 7, both aspects of vulnerability assessment (Detection and Analysis) and mitigation (Containment). Our paper is also a foundational work of a novel research direction towards the automatic and preventative elimination of security vulnerabilities in networked systems controlled by SDN. Thus, the current survey enriches available literature.

III. PROACTIVE DETECTION OF SECURITY VULNERABILITIES

In the previous section, the background to this paper and its relevance were presented, as follows: i) explaining the paper motivation and context; ii) writing about relevant foundational concepts such as vulnerability detection, classification, and mitigation as well as solutions design based on SDN; and iii) highlighting the novelty of our paper in relation to previous related surveys. We should be aware of that considering the new APIs and associated protocols imposed by typical SDN design, the potential attack surface increases, requiring innovative and more efficient approaches to proactively detect system vulnerabilities.

Referring now to the current section, the Table 5 lists the main characteristics of the surveyed work for detecting vulnerabilities using the system control level. This table contains a set of comparison parameters, as follows:

- Vulnerability Assessment: if the analyzed work performs vulnerability assessment, i.e. detecting and/or mitigating vulnerabilities. This parameter is used to highlight papers that are more concerned with the proactive attack prevention rather than the reactive detection of an ongoing attack.
- SDN Controller: SDN Controller is used or supported.
- Automation: the work proposes automatic tasks with minimal human intervention, aiming to streamline processes, reduce errors, and enhance productivity.
- Risk Indicator: The work classifies the risk represented by the anomaly. CVSS implies that the work used the Common Vulnerability Scoring System; Custom is referred when the work used its own risk classification model. The work may also have used CVSS as a basis in conjunction with custom metrics to enhance the anomaly classification.
- Passive Scanning: Defined in II-D. Analyzes traffic passing through a network monitoring point.
- Active Probing: Defined in II-D. Interacts with services by sending packets to each host and monitoring their response.
- Proximity Score: this value can vary between 0 and 20, in an attempt to show how close the work is to the comparison parameters. This value is not intended to assess the quality of the work.

The most of the surveyed work adopted and tested a solution to automatically detect the system vulnerabilities, using a SDN controller. Nevertheless, only a few [39], [40],

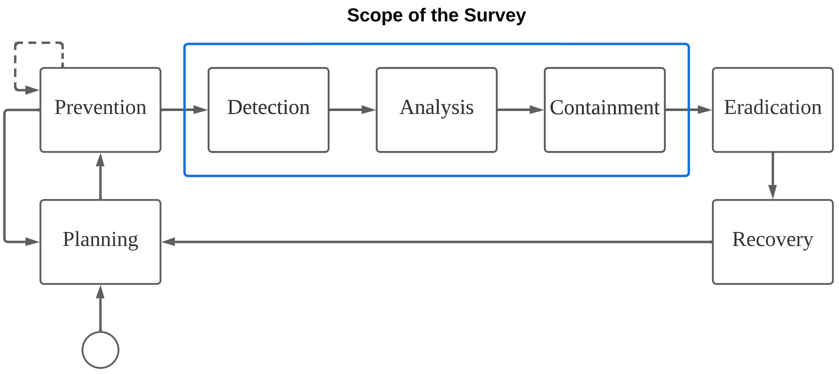

FIGURE 7. Scope of this survey.

TABLE 5. Detection papers comparison.

Paper	Vulnerability Assessment	SDN Controller	Automation	Risk Indicator	Passive Scanning	Active Probing	Proximity Score
[39]	●	POX	●	Custom	○	●	16.7
[40]	●	ODL, ONOS	◑	CVSS, Custom	○	●	15.0
[41]	●	ONOS	●	CVSS	○	●	16.7
[42]	●	ODL	●	Custom	●	○	16.7
[43]	○	N/A	●	○	●	○	6.7
[44]	○	ODL	●	○	●	○	10.0
[45]	○	N/A	●	○	●	●	10.0
[46]	○	N/A	●	Custom	●	○	10.0

Table Guide: ●: when the work satisfies the parameter analyzed and the authors make a direct or indirect mention of it. ◑: when the work does not directly mention the designated parameter, but the work seems to partially comply with it. ○: when the work does not meet the designated parameter. N/A: when for the SDN Controller column, the authors do not mention the controller used/supported. This table guide also applies for next Tables 6 and 7.

[41], [42], [44] have clearly identified the controller used or supported. In addition, from the analyzed contributions, the passive scanning [42], [43], [44], [45], [46] was by far more popular than the active scanning [39], [40], [41], [45] of system vulnerabilities. Further, the incorporation of a risk indicator to assess the system vulnerability level was marginal [39], [40], [41], [42], [46] among the surveyed references and, among these, only [40] and [41] adopted a standard risk metric. From our analysis, there is a clear opportunity to strength future research on novel active scanning techniques for discovering system vulnerabilities, using well-known SDN controllers and standard system risk metrics. References [39], [40], and [41] deserve to be recognized for their quality and relevance to what we intend to investigate in the current survey. Further details are given in below text.

The work done at [39] focuses on implementing a system that detects vulnerable IoT devices and attempts to fix their vulnerabilities before being accepted into the network. A POX Controller, firewall, DHCP server and a Host Tracker were used. A new arriving device sends a IP request to the DHCP server which responds and starts a vulnerability scan to inspect the device. Two tools are used to scan for vulnerabilities, NESSUS and a custom weak password scanner. Following the scan result, the new device can be added to one of two lists: whitelisted or blacklisted. Whitelisted devices correspond to devices in which no vulnerabilities have been found. Alternatively, blacklisted devices are considered vulnerable and dangerous to the system security. In addition, for every packet sent, the Host Tracker component checks if both source and destination hosts have been already scanned. This does not seem a good option, because it forces the controller intensively work in reactive mode, increasing the traffic latency.

Paper [40] proposes a systematic approach to evaluate and optimize the security posture of SDN and emphasizes the importance of analyzing the effectiveness of countermeasures in mitigating various threats faced in SDN. It introduces a framework for threat modeling and security assessment, utilizing three security metrics: network centrality measure, vulnerability score, and attack impact metrics. They have also developed a novel graphical security formally designated as Threat model using Threat Vector Hierarchical Attack Representation Mode (TV-HARM). TV-HARM enables security risk assessment of the SDN system. Experimental analysis was conducted to demonstrate the applicability of the framework and TV-HARM in capturing various threat vectors in SDN. The paper provides insights into the potential new threats in SDN and offers a comprehensive approach for evaluating the security of SDN.

Vulnerability Assessment as a Service (VAaaS) [41] cross-layered system is divided into three layers: Private Cloud, Fog, and Extreme Edge. In the Private Cloud, a Kubernetes container orchestrator manages the infrastructure, namely a primary ONOS SDN Controller. This SDN controller communicates with a monitoring service that communicates with the Decision Engine that takes different decisions depending on the reported event. The decision engine starts the assessment procedure by requesting Kubernetes to deploy an instance of OpenVAS to the below Fog layer Kubernetes orchestrator associated to the reported device event. A registry of the underlying hosts and their assessment status is kept locally in a MongoDB database which includes their cybersecurity status, CVSS score, certification timestamp, VLAN identification, and other valuable information. In the Fog layer there are several Fog nodes. The Extreme Edge is the system's lowest abstraction layer, where all devices, virtual and real, are installed/deployed. The monitoring service will receive the controller's list of connected hosts through its northbound API and send it to the Decision Engine which checks the MongoDB database and if there is a device in the registry that is not certified, the certification process is initiated by instructing the monitoring service to assign each device to a neutral, limited-connectivity-VLAN, and the orchestrator to deploy the OpenVAS-based assessment agent. As a result, it informs the deployed OpenVAS agent to analyze all unassessed devices. Until the assessment is completed, the devices remain members of the limited-connectivity VLAN. The vulnerability assessment generates a score for each device that is based on the CVSS which is used to assign each device to a connectivity-appropriate VLAN. As a result, if a device's evaluated severity is "None", it will be allocated to the full-access VLAN. If the severity of the device is between "Low" and "Medium", it will be allocated to the restricted-access VLAN. Similarly, if the evaluated severity of the device is "High", it will be allocated to the no-access VLAN. Finally, after completing its task, the orchestrator destroys the assessment mechanism. Every new device connected to the network follows the same procedure. Furthermore, the procedure is performed on a regular basis for all connected devices, every 10 days (pre-configured value).

DIVERGENCE [42] aims to provide scalable and intensive network traffic visibility for rapid threat detection and defense. The framework includes two main security services: a DRL-based network traffic inspection mechanism and an address shuffling-based MTD technique. By utilizing DRL, DIVERGENCE learns an optimal traffic inspection resource allocation policy under the uncertainty of malicious flow occurrence and performs MTD according to traffic inspection results reported from multiple IDSs.

Research [43] tries to improve the performance of an IDS system in SDN by identifying and preventing attacks on SDN devices. The proposed framework utilizes techniques based on signatures, an anomaly-based IDS for evaluating patterns of traffic and detects threats.

Study [44] examines the identification and prevention of attacks by utilizing RL techniques inside SDN. An agent modifies network security settings according to the current state of the environment while receiving a curiosity incentive signal that encourages discovery. The framework was assessed by subjecting it to attack scenarios using selected datasets.

SDNRecon [45] assesses the efficacy of cyber deception tactics, with emphasis on collecting sensitive data that may assist attackers in carrying out further harmful actions. The tool evaluates many elements of SDN networks, such as identifying the controller vendor, retrieving host information, and discovering vulnerabilities. SDNRecon emphasizes the importance of reconnaissance in the attackers' process of obtaining information, praising the efficacy of the SDNRecon tool in evaluating and improving cyber deception techniques. It also emphasizes the comparison with alternative technologies and the potential for synergy between reconnaissance tools and cyber deception systems.

Article [46] presents a methodology that examines attacks and generates risk assessment scores. It uses the Generalized Stochastic Petri Net (GSPN) model to analyze DoS attacks. The findings offer insights into attack paths, and investigate the correlation between risk levels and the timing of attacks. The results emphasize the clear relationship between the likelihood of risk and the average duration of attacks, offering useful insights for enhancing security evaluations and developing efficient countermeasures. The work hasn't been tested.

One of the weaknesses of the literature is the extremely small number of studies that perform vulnerability assessments which is crucial for firms seeking to strengthen their security defenses and actively mitigate possible threats, and this way avert security breaches and data leaks. Furthermore, vulnerability assessments enhance strategic decision-making by offering valuable insights into new threats and directing investments in security. It is therefore crucial that more research into this matter should be done. We would like to see greater importance given to the integration of active probing tools in SDN since they allow organizations to promptly detect vulnerabilities by actively examining their systems, uncovering potential ports of entry and flaws. They facilitate the prioritizing of mitigation operations, enabling companies to rapidly and efficiently address the most crucial system vulnerabilities. This methodology improves the readiness for responding to incidents, since the knowledge obtained from active scanning informs about the creation and improvement of proactive measures against such incidents.

IV. PROACTIVE ENHANCEMENT ON SYSTEM SECURITY

In the previous section we looked at vulnerability/threat detection technologies. The proactive enhancement on system security is critical to reduce the risk of future threats putting at risk the normal operation of programmable networked systems. As the number of threats and their sophistication increases, it is essential to implement proactive measures to help organizations mitigate potential system vulnerabilities before they can be exploited by malicious actors. To attain this, various proactive techniques can be

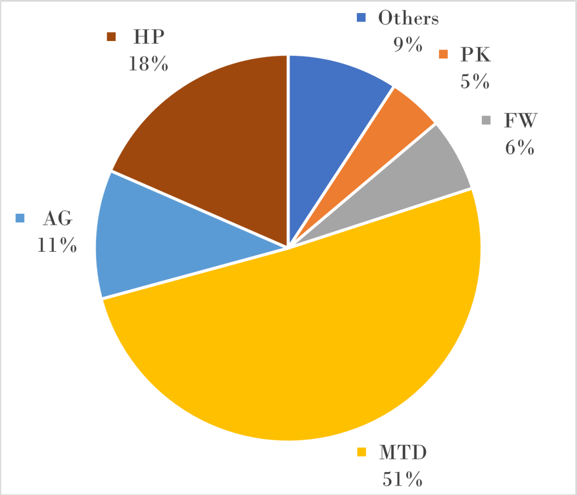

FIGURE 8. Percentage of mitigation methods employed.

deployed to identify, assess, and mitigate to possible attacks in real-time. Deceptive technologies such as MTD and honeypots can be used to build a virtual environment designed to mislead attackers away from critical systems and private data. Fig. 8 shows the percentage of mitigation proposal types studied in the revised literature. As we can see, MTD is by far the mitigation technique most covered in the literature (50%), followed by HP with a much lower percentage (19%). Although we don't consider AGs to be a mitigation technique, but rather a tool to support it. We thought it relevant to consider it for Fig. 8 and for Tables 6 and 7, as AG enhances the approaches capability to classify the discovered vulnerability risks.

Tables 6 and 7 list work focused respectively in the data and control planes, using the next comparison parameters:

- Technique: the used mitigation technique to reduce the risk created by threats. These techniques are taxonomized in Fig. 6, with the exception of CNN.
- SDN Controller: SDN Controller is used or supported.
- Automation: the work proposes automatic tasks with minimal human intervention, aiming to streamline processes, reduce errors, and enhance productivity.
- Elasticity: we consider the proposed system to be elastic, in other words capable of adapt to real-time varying conditions. This parameter assumes the work fulfills one of three elasticity characteristics: scalability, flexibility/adaptability or dynamic resource allocation.
- Risk Indicator: the work classifies the risk posed by the anomaly.
- Latency: the work proposal has impact on the network traffic delay
- Throughput: the work proposal has impact on the network throughput.
- Proximity Score: this value can vary between 0 and 20, in an attempt to show how close the work is to the comparison parameters. This value is not intended to assess the quality of the work.

A. DATA PLANE

The current survey explores the technologies used to strengthen the data plane, guaranteeing the safe and efficient transfer of data while actively reducing vulnerabilities. Proactive measures implemented in the data plane play a crucial role in constructing robust and reliable network infrastructures. More than half of the studies analyzed performance in terms of latency [47], [48], [49], [50], but only one looked at the impact on throughput [47]. The methods used were Convolutional Neural Networks (CNN) [47], Moving Target Defense (MTD) [48], [49], Port Knocking (PK) [50], [51], [52] and Firewall (FW) [47], [52], [53]. None of the studies used risk indicators and only [53] mentioned the used SDN Controller. The [47], [51], and [53] papers tried to replace manual tasks with automated ones, while the [47], [49] and [53] contributions evidenced an elastic capability. In the below text, we detail the discussion on the selected contributions.

Article [47] focuses on the security risks associated with IoT, highlighting the importance of using firewalls, SDN, and the P4 language for detecting attacks. It proposes a two-stage deep learning method for creating flow rules, which is evaluated and shown to outperform existing approaches. The contributions encompass a pioneering architecture for safeguarding IoT networks and devices, offering a effective resolution for detecting malevolent data streams.

Investigation [48] presents an MTD method executed at the data plane and on every node along the forwarding path. The approach increases the cost for attackers to carry out network reconnaissance by randomizing network addresses. This protection is implemented at the link layer, ensuring secure access at both router switches and end-hosts. The solution additionally tackles a sophisticated threat scenario involving compromised network nodes, which disseminate controller communication information to provide MTD randomization. The results indicate a rise in the cost that attackers need to spend on timely reconnaissance, establishing the scheme as an efficient method for ensuring secure access to forwarding paths in SDN. The authors extend their work in [49] where they incorporate the randomization of IP addresses for MTD and use them for synchronization among network nodes, making use of the already existing IP addresses for randomization. This leads to a modular solution that is not dependent on the implementation of routing or flow rules, and it also causes little additional networking cost. The scheme's efficacy is shown by significantly raising the cost of network reconnaissance for attackers by more than ten times compared to the prior literature. The scheme is also capable of being scaled up, with the proportionate additional cost diminishing as the network expands. The article conducts a comprehensive examination and verification of the system by implementing and experimenting on a testbed based on OpenvSwitch and CloudLab.

TABLE 6. Mitigation papers comparison - data plane.

Paper	Technique	SDN Controller	Automation	Elasticity	Risk Indicator	Latency	Throughput	Proximity Score
[47]	CNN, FW	N/A	●	●	○	●	●	14.3
[48]	MTD	N/A	○	◑	○	●	○	7.1
[49]	MTD	N/A	○	●	○	●	○	8.6
[50]	PK	N/A	◑	◑	○	●	○	8.6
[51]	PK	N/A	●	○	○	○	○	5.71
[52]	FW, PK	N/A	○	○	○	○	○	2.9
[53]	FW	POX	●	●	○	◑	○	12.9

P4Knocking [50], a PK-based authentication mechanism implemented in the P4 language that offloads host-based authentication functionality to the network. The implementation relied on registers that hold values that work like counters to track the state of the knock sequence for a given source IP address where these counter defines the access to a given destination IP address. The implementations focused on offloading the host-based authentication functionality to the network and making the mechanism transparent for the end host. It concludes P4Knocking to be more transparent and efficient compared to a host-based PK implementation.

PortSec [51] indicates the susceptibility of conventional PK sequences due to their static nature and introduces three new communication protocols based on sequences: static, partial dynamic, and dynamic. Each protocol provides a greater level of security than the previous. These protocols are specifically intended to operate inside the data plane.

P4Filter [52] leverages P4 to enhance network security through a two-level defensive approach. The first level of defense is a dynamic firewall that incorporates both stateful and stateless firewall concepts, effectively blocking packets from unauthorized sources. The second level of defense is an authentication mechanism that employs PK. The P4Filter's packet processing involves three main modules within the P4 switch: two security modules for filtering packets and one forwarding module. When a packet arrives at the switch from an unknown host, it is sent to the controller, which maintains an ACL. The controller uses this ACL to assign rules that determine whether to allow or drop packets based on the security levels. If a packet does not match the internal network's criteria, a 'direction' bit is set, and a hash is calculated using various packet attributes.

Article [53] presents a stateful firewall intended for cloud environments. This FW differs from standard cloud FWs in terms of its approach to security rule configuration and rule matching, which are typically static and basic. The FW utilizes a finite state machine and a state table to directly extract, analyze, and record the connection status information of data packets within the data plane. The data plane packet processing is specifically built to execute stateful inspection and integrity checks by parsing and analyzing the structural information of the packet header.

Data plane mitigation papers lack of risk indicators to accurately assess risks. This poses as a notable hole as it impedes security measures to prioritize the most crucial vulnerabilities or hostile behaviors, resulting in a less efficient response to potential threats. Taking advantage of the capabilities of the data plane offers great potential, especially in tackling issues related to latency and resource allocation. The data plane, offers an interesting chance to enhance latency by moving security measures closer to the system hardware. This allows for rapid processing and decision-making, reducing the time delays often associated with conventional security methods. Furthermore, utilizing the data plane to distribute security tasks among network devices enables an efficient distribution of tasks, improving scalability and adaptability. The adoption of this decentralized strategy not only enhances the overall efficiency of security measures but also enables robust and adaptable security solutions running at the hardware speed.

B. CONTROL

This subsection explores mitigation solutions that were implemented at the control plane which help in preventing unauthorized access, minimizing the risk of attacks, and maintaining the overall integrity of the network. Fig. 9 shows the percentage of controller options mentioned in the various mitigation and detection works, from which ODL and Ryu are the most popular controller options in the revised literature.

From Table 7 the most researched mitigation technique is MTD [22], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [85], [86]. Honeypot (HP) is also a well-researched technique [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92]. Both of these deception techniques can be combined with AGs, as shown in [22], [74], [75], [76], and [77] for MTD and [82] for HP, but they can also be used in conjunction with DPI [93]. Other access control techniques such as PaH [79], PoH [80] or FW [39] are also present. SDN controllers Ryu [63], [64], [67], [68], [69], [81], [85], [90] and ODL [70], [72], [73], [74], [83], [89] are the most widely used/supported in the surveyed control plane literature. The concern to replace manual processes with automated strategies is present in papers [39], [63], [65], [66], [74], [75], [76], [81], [86], [87], [88], [89], [90], [92], [93]. Roughly half of the systems show characteristics of elasticity [22], [54], [56], [60], [66], [70], [73], [74], [75], [76], [78], [81], [82], [83], [84], [86], [87], [89], [91], [93]. There weren't found many papers to use risk indicators except [22], [54], [61], [66], [70], [74], [75], [76], [77], [82], [87], [93] and from these only [22], [74],

TABLE 7. Mitigation papers comparison - control plane.

Paper	Technique	SDN Controller	Automation	Elasticity	Risk Indicator	Latency	Throughtput	Proximity Score
[54]	MTD	N/A	○	●	Custom	○	○	8.6
[55]	MTD	N/A	○	○	○	●	●	8.6
[56]	MTD	N/A	○	●	○	○	○	5.7
[57]–[59]	MTD	ONOS	○	○	○	○	○	5.7
[60]	MTD	N/A	○	●	○	●	●	11.4
[61]	MTD	N/A	○	○	Custom	○	○	5.7
[62]	MTD	ONOS	◑	◑	○	◑	○	10.0
[63]	MTD	Ryu	●	○	○	●	●	14.3
[64]	MTD	Ryu	○	◑	○	●	○	10.0
[65]	MTD	N/A	●	○	○	◑	○	7.1
[66]	MTD	N/A	●	●	Custom	◑	◑	14.3
[67]–[69]	MTD	Ryu	○	○	○	○	○	5.7
[70]	MTD	ODL	◑	●	Custom	○	○	12.9
[71]	MTD	N/A	○	○	○	○	○	2.9
[72]	MTD	ODL	◑	◑	○	○	○	8.6
[73]	MTD	ODL	○	●	○	●	○	11.4
[74]	AG, MTD, PoH	ODL	●	●	CVSS, Custom	○	○	14.3
[75]	AG, MTD	N/A	●	●	CVSS, Custom	○	○	11.4
[22]	AG, MTD	N/A	○	●	CVSS, Custom	●	○	11.4
[76]	AG, MTD	N/A	●	●	CVSS, Custom	●	○	14.3
[77]	AG, MTD	N/A	◑	◑	CVSS, Custom	○	○	8.6
[78]	CNN, MTD	N/A	○	●	○	●	○	8.6
[79]	MTD, PaH	N/A	○	○	○	●	○	5.7
[80]	MTD, PoH	NOX	○	○	○	○	◑	7.1
[81]	HP	Ryu	●	●	○	○	○	11.4
[82]	AG, HP	N/A	○	●	Custom	○	○	8.6
[83]	HP	ODL	◑	●	○	○	○	10.0
[84]	HP	N/A	◑	●	○	●	●	12.9
[85]	HP, MTD	Ryu	○	○	○	◑	○	7.1
[86]	HP, MTD	POX	●	●	○	◑	◑	14.3
[87]	HP	N/A	●	●	Custom	○	○	14.3
[88]	HP	N/A	●	◑	○	○	○	7.1
[89]	HP	ODL	●	●	○	○	○	11.4
[90]	HP	Ryu	●	◑	○	○	○	10.0
[91]	HP	FL	◑	●	○	●	○	12.9
[92]	HP	ONOS	●	○	○	●	○	11.4
[39]	FW	N/A	●	◑	○	○	○	7.1
[93]	AG, DPI	N/A	●	●	CVSS, Custom	○	●	14.3

[75], [76], [77], [93] used a standard risk metric. In terms of performance impact, works [22], [55], [60], [63], [64], [73], [76], [78], [79], [84], [91], [92] studied traffic latency but only [55], [60], [63], [84], [93] examined the impact on throughput. Honorable mention to [22], [39], [74], [75], [76], [84], and [93] for their proximity to the main theme of this review. We give more details in the text below.

1) MOVING TARGET DEFENSE

SDN-oriented Cost-effective Edge-based MTD Approach (SCEMA) [54] and [55] present MTD approaches to protect SDNs against DDoS attacks. This is accomplished on SCEMA [54] by shuffling a well-tuned group of hosts that have strong connections to important servers while [55] uses virtualized addresses for end-hosts, regularly remapping virtual IP addresses, hiding the actual ones.

Paper [56] examines the failure rate of algorithms in relation to mutation cost and trusted resource when some of these resources have been already under network attacks. The algorithm successfully mitigates attacks on nodes and ensures a decreased rate of service failures.

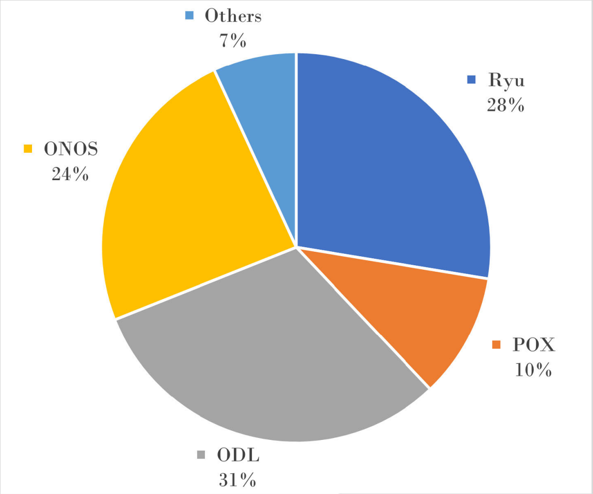

FIGURE 9. SDN controllers mentioned in the revised literature.

Study [57] examines the practical efficacy of a virtual IP shuffling MTD approach in an SDN testbed. It takes into account two categories of attackers: dummy and adjusting.

The results confirm the effectiveness to prevent attacks in a practical network environment. It also states the need of continuous adaptation and assessing security in the presence of different levels of attacker knowledge.

MTD Adaptive Delay System (MADS) [60] provides efficient protection against scanning attacks. MADS employs a selective activation of the adaptive delay mechanism, where delays are only applied to response packets upon detection of an attack. MADS exhibits reduced network degradation in several aspects, including latency and throughput. The conclusion recognizes the effectiveness of MTD approaches in protecting SDN, while also emphasizing the potential negative impact on network performance. FRVM (Flexible Random Virtual IP Multiplexing) [61], aims to defend against reconnaissance and scanning attacks by enabling hosts to have several virtual IP addresses that are randomly allocated and changed over time. The multiplexing event guarantees a significant level of network diversity among hosts. FRVM is used by [62] that employs multiple SDN controllers to enhance both security and performance in large-scale networks, reducing the problems of having a single point of failure and scalability limitation. Paper [63] evaluates the effectiveness of MTD as a scan countermeasure, and its ability to detect low and high-rate scans. Results indicate that the combined use of MTD and IPS is effective in countering various rates of scans, and the proposed approach minimally affects communication delay and throughput.

SDN based Moving Target Defense for Control and Data planes Security (SMCDS) [59] framework is designed to protect control and data planes from reconnaissance attacks. It leverages distributed shadow controllers to secure the control plane, enhancing its resilience and availability by presenting attackers with a constantly changing target. For the data plane, SMCDS combines reactive and proactive strategies, using shadow servers to deflect reconnaissance traffic and shuffling IP and port addresses. Theoretical models for calculating attacker and defender success probabilities are provided, and experimental results validate the defender's success across various scenarios.

Dynamic Random Route Mutation (DRRM) [64] mechanism is presented as a solution for MTD. It increases the randomness of mutations, minimizing the time required to perform them. The approach employs the Jaccard distance matrix and temporal restriction to enhance the mutation space, consequently diminishing the eavesdropping capabilities of attackers at certain nodes. In addition, a pre-distribution approach is employed to minimize transmission delay resulting from route mutation.

Methodology [65] employs a ML-based model to categorize rules and detect various attack types from malevolent network traffic. The paper outlines the design and execution of the suggested hybrid MTD system, and assesses its security benefits.

Frequency-Minimal Moving Target Defense (FM-MTD) [66] aims to minimize resource waste and loss of availability while effectively countering attackers. It achieves this by dynamically MTD across heterogeneous VMs based on attack probability. The scheme considers factors such as VM capacity, network bandwidth, and VM reputation to identify the ideal VM for migration. The results show that the FM-MTD scheme outperforms static schemes in terms of attack success rate.

The article [67] presents an experimental setup using Mininet and Ryu controller, and discusses the implementation of some MTD techniques. It also delves into the analysis of TCP and UDP traffic in both traditional and MTD SDN network topologies, revealing considerable overhead on the controller, resulting in poor performance on several important network features, such as latency, jitter, and packet loss. Findings emphasize the need for further research to minimize these system overheads. Further work [68] from these authors, indicate the high pertinence on future work for reducing the controller overhead when the MTD technique is used.

The suggested MTD approach in [69] involves the mutation of IP addresses inside SDN. The effectiveness of the MTD approach is demonstrated through experimental assessment utilizing the Ryu controller and Mininet. IP mutation is identified as a viable method to incorporate MTD into SDN systems and reduce the risks of poisoning attacks. The main goal of a poisoning attack is the attacker to collect relevant initial information about the network operational states, to initiate DDoS attacks later more easily.

The article [70] introduce the Shuffle Assignment Problem (SAP), a complex problem formulated to reconfigure network topology, which scales exponentially with network size. This mechanism is further refined by the introduction of a topological distance metric, which aids in selecting the most effective countermeasures in real-time when an attack is detected. The Expected Path Value (EPV) is a critical metric used within the study to evaluate the effectiveness of network reconfigurations by quantifying the probability of an attacker reaching a target through compromised paths. The Defensive Network Reconfiguration Scenarios (DNRS) are a set of pre-computed network topologies, each with an associated EPV, from which an optimal topology can be selected to counter detected threats.

POSTER [71] highlights that the increasing prevalence of AI is leading to increasingly advanced attacks, necessitating a re-evaluation of the effectiveness of MTD. The researchers developed a classification system for possible attacks against MTDs. The framework is intended to facilitate the creation of datasets that can be employed to replicate these sophisticated attacks. The paper points out a form of attack where an attacker may use ML methods to analyze network data and determine the MTD interval for a time-based MTD. Thus, an attacker can initiate the attack right after the last MTD trigger, maximizing the time to complete the attack. To increase the difficulty on the attacker having success in his/her reconnaissance attack phase, the authors propose a system defense technique like adding noise to the network data for hiding the MTD trigger from the attacker.

Random Host and Service Multiplexing (RHSM) [72] introduces a dynamic shuffling mechanism for IP addresses and port numbers to obfuscate the real identities of hosts and services at both the network and transport layers. RHSM employs a multiplexing strategy that allows each host to use multiple virtual IP addresses and services to use multiple virtual port numbers, which are periodically and dynamically shuffled. This shuffling is managed by a proxy installed on each end-host, which performs the necessary translations between virtual and real IPs and ports for incoming and outgoing packets. The article demonstrates through simulation experiments that RHSM significantly reduces the attack success probability (ASP) when compared to a static network configuration, while also considering the defense cost.

MTDSynth [73] objective is to improve agility by allowing systems to actively protect themselves against advanced threats through the dynamic adjustment of system configurations. The paper showcases the practicality of the framework by providing examples of temporal and spatial IP mutation, route mutation, and detector mutation implemented using ActivSDN on the ODL SDN controller.

a: ATTACK GRAPHS

MASON [74] introduces a threat score system that relies on vulnerability information and intrusion attempts to identify high-risk VMs. The MTD countermeasures, specifically port hopping, are then implemented to evaluate the reduction in threat scores within the cloud network. The study includes experiments that analyze network threat scores based on software vulnerabilities and IDS alerts, which help prioritize the most vulnerable services for MTD countermeasures. Additionally, the research examines the effectiveness of MTD countermeasures in relation to the attacker's reward, considering varying numbers of services and VMs. The findings indicate that as the number of nodes or services on the network increases, so does the threat score, highlighting the increased complexity and potential attack paths in larger networks.

In reference [75], MTD incorporates the use of AGs. The analytical models employed in this study integrate both random and weakest-first attack behaviors, effectively reflecting the preferences of attackers as well as the topological aspects of the system under analysis. The MTD system under consideration employs Bayesian Attack Graph (BAG) analysis as a means to evaluate security concerns and provide guidance for adaptive MTD operations. This approach is essential in the estimation of ASP for vulnerabilities, taking into account various elements like topological degrees and attacker scanning habits.

Framework [22] emphasizes the rearrangement of network configurations for hosts that are both critical and vulnerable. The frequency of this rearrangement is defined by the level of criticality assigned to each host. The assessment of host exploitability is based on severity rankings assigned to vulnerabilities and the probability of successful attacks. The system includes the Three-tier Attack Graph (TAG) graphical model, which simplifies the analysis of host exploitability. In addition, a mechanism for predicting attack paths has been developed to safeguard important assets, and a method for managing system performance is proposed, which allows for control over address shuffling. The findings emphasize the efficacy of the asset in hiding network information from attackers and its ability to offer scalable and adaptive security, while maintaining an acceptable level of computation cost and latency.

Paper [76] presents an integrated defense mechanism designed for IoT environments, with a focus on resource-constrained devices that are highly susceptible to attacks. The authors propose a combination of deception techniques, which involves the use of decoy nodes, and MTD. The authors develop and analyze four strategies for determining ''when'' to shuffle the network topology—fixed, random, adaptive, and hybrid and three strategies for ''how'' to perform the shuffling—genetic algorithm, decoy attack path-based optimization, and random. The study is conducted in the context of a smart hospital scenario, but the approach is applicable to any IoT environment. The results demonstrate that the proposed technique can extend the system's lifetime, increase the complexity of attacks on critical nodes, and maintain high service availability compared to IoT networks without this defense mechanism. Additionally, the authors provide insights into the best combination of ''when'' and ''how'' strategies to achieve specific system goals, such as maximizing system lifetime and service availability while minimizing defense costs.

Article [77] proposes a proactive defense mechanism for IoT networks by combining MTD with cyber deception through the use of decoy nodes. The proposed framework introduces security metrics such as Attack Cost (AC), Return on Attack (RoA), and Risk (R) from both attacker and network defense perspectives. The authors have also provided a technique for reducing the cost associated to system defense.

b: CONVOLUTIONAL NEURAL NETWORKS

AHIP [78], is an adaptive IP hopping technique designed for MTD, intending to mitigate a range of network threats. By employing a lightweight one-dimensional convolutional neural network (1D-CNN) detector, AHIP can dynamically choose IP hopping techniques according to network conditions, therefore efficiently mitigating scanning and DoS attacks. The suggested approach is implemented using SDN, where the SDN controller is responsible for installing the 1D-CNN detector and the IP hopping module. The experimental findings conducted within a simulated environment provide evidence of AHIP's efficacy in safeguarding against network threats, while concurrently decreasing the burden on the system. The activation of adaptive IP hopping tactics is contingent upon the output of the trained 1D-CNN detector, hence facilitating AHIP's ability to successfully counter various attack scenarios.

c: PATH/PORT HOPPING

Path Hopping Based SDN Network Defense Technology (PH-SND) [79] presents a technology known as Path Hopping which is designed to improve the network's defense by dynamically altering the paths that data packets take through the network. PH-SND approach involves modeling the path hopping problem as a constraint-solving issue, where the goal is to find multiple paths that meet specific overlap and capacity constraints. Once suitable paths are determined, the controller installs the necessary flow entries into all switches along each path. These switches are then responsible for forwarding the protected flow and are capable of randomly changing the address and port information of the flow to further obscure the communication details between the sender and receiver.

Port Hopping technique with Masked Communication Services (PHCSS) [80], addresses the limitations of service port masking and the need for additional hardware, which typically result in increased overhead. The proposed technique is designed to efficiently detect and filter malicious packets, thereby reducing the server's port hopping costs and resisting against various network attacks. The experimental setup utilized Mininet, Open vSwitch, and NOX controller to simulate an SDN network environment, demonstrating that PHCSS can secure a network from port scanning and DoS attacks without overburdening the SDN controller's resources.

2) HONEYPOTS AND HONEYNETS

Article [81] describes a honeynet system that captures and monitors incoming traffic to identify and gather data on malicious attacks and the behavior of the attackers. The system employs a combination of physical and virtual HPs, with SDN and container technologies, to facilitate the dynamic generation of honeynets and strengthen their deceptive capacity.

Paper [82] presents a deception resources allocation model based on SDN, which incorporates a multi-layer AG and a signaling game. This allows for the dynamic allocation of deception resources according to the severity of the threats. The system integrates geographical data into a multi-layer AG and offers a Top-N module for filtering attack paths. In addition, the model utilizes a signaling game strategy to determine flow scheduling, which is measured by the AG.

Dynamic Virtual Network Honeypot (DVNH) [83] employs dynamic instantiation of honeypot systems to efficiently divert attacks and protect targeted systems. It tackles the conventional issues of deploying HPs and managing costs by dynamically adjusting capacity according to demand while simplifying the operational complexity linked to honeypot administration.

HONEYPROXY [84] provides a flexible network access management system, globally monitoring internal traffic and supporting dynamic transitions between low-interaction and high-interaction HPs. By utilizing SDN, it enhances data control and capture capabilities, preventing fingerprinting attacks and improving overall resilience. The proposed architecture redistributes malicious traffic to HPs, allowing response selection without relying on fingerprinting indicators. Experimental results demonstrate HONEYPROXY's high throughput and minimal latency overhead, establishing it as an effective solution for advanced honeynet functionality.

The authors introduce a combination of MTD and SDN-based honeypots in [85]. The MTD architecture involves constantly changing the IP addresses of IoT devices and servers. Additionally, SDN-based honeypots are deployed to mimic IoT devices, luring attackers and malware. Experimental results demonstrate the effectiveness of this approach in defending against DDoS attacks and hiding network assets from malicious scanning.

MTD Enhanced Cyber Deception protection System (MTDCD) [86] offers protection mechanism against Advanced Persistent Threat (APT) attackers, which are commonly initialized via network reconnaissance. The method utilizes IP address randomization and the system's architecture consists of three primary components: the module for virtual network topology, the module for IP randomization, and the deception server. The primary role of the deception server is to detect and counteract malicious scanners while maintaining the appearance of an authentic network. The implementation of the MTDCD system resulted in a seven fold increase on the time for adversaries to identify susceptible hosts, and it also decreasing the probability of successful attacks by 83%.

HoneyV [87] utilizes multi-phase data monitoring to improve the detection of malicious activities. Instead of terminating all sessions deemed suspicious, HoneyV dynamically routes traffic to server replicas with varying levels of monitoring intensity, depending on the assessed risk of attack. By doing so, HoneyV provides IDSs with training capability.

Paper [88] proposes a honeypot system that utilizes the combination of SDN and Recursively Defined Topologies (RDT). The authors present a mathematical approach to describe RDTs and propose an algorithm for its generation. The objective is to develop efficient honeypots that can simulate complex data center environments on a single physical host. The design combines SDN with an orchestrator engine to create containerized infrastructure, which enables the creation of high-interaction honeypots. The work hasn't been tested.

S-Pot [89] is a smart honeypot framework with dynamic flow rule configuration for SDN. The authors conducted a performance evaluation of S-Pot in an enterprise SDN testbed network, simulating various types of attacks. The results demonstrated that S-Pot could detect attacks with a high accuracy of 97%. Furthermore, the study showed that S-Pot could improve the security of SDN networks by effectively generating rules and dynamically configuring the network, leading to better performance and greater accuracy.

The article [90] presents a network model for an intelligent honeynet. The proposed model is structured into three layers:

the infrastructure layer, which includes network and honeynet exchange equipment; the controller layer, which interacts with the infrastructure layer via the SBI; and the application layer, which develops specific applications using the NBI API provided by the control layer. The intelligent honeynet comprises the attack migration mechanism, the topology management mechanism, the attack detection module, the policy generation module, and the honeynet management module. The attack migration mechanism involves the detection of attacks and the generation of strategies for traffic forwarding. The topology management mechanism dynamically generates honeynet nodes, links, and routing information to adapt to ongoing attacks. The article validates the model through experiments using Mininet to simulate attacks and demonstrate the honeynet's performance.

The paper [91] intends to overcome the obstacles of flow management and topology modeling. The suggested solution leverages the scalability and manageability of the SDN controller to emulate intricate network structures and stealthily redirect malicious data from a basic interface to a more advanced interface for extensive analysis. The design consists of a topology management module that maintains virtual topology information, an ARP simulation that handles ARP queries, and a flow table lifecycle management module that manages the metadata of flow tables and ensures their effective timing in the OpenFlow switch. The system incorporates a method for migrating attack traffic, which categorizes attacks and redirects them to suitable honeypots according to their degree of complexity.

SDNHive [92] aims to counter the spread of ransomware within a network. SDNHive utilizes the capabilities of the SDN controller to perform intrusion prevention measures like address blacklisting, connection blocking, and transparent traffic rerouting. The system includes a honeypot that functions as an active intrusion detection device.

3) OTHER METHODS

One of the interesting things about the proposed system in [39] is that for devices for which vulnerabilities have been found the system will try to resolve them. If successful, the device will be allowed to join the network (whitelisted). If the vulnerabilities remain, the device will be blacklisted and an email will be sent to the device user offering suggestions to fix them. The decisions made by the scan server are translated into an ACL managed by the DHCP server, which is used by the firewall to enforce security policies and restrict access to the network accordingly. The firewall inserts rules in the SDN controller which in turn updates the flow tables of the OpenFlow switch.

Network Intrusion detection and Countermeasure sElection (NICE) [93] uses AG Analytical procedures and allows the cloud to inspect and isolate suspicious machines (VMs) according to the current state of Scenario Attack Graph (SAG) which is defined by parameters such as IP addresses, vulnerability information (e.g. CVE) and alarmistic data. There is an agent called NICE-A in each cloud server that performs periodical vulnerability scans in VMs. Based on the severity of each vulnerability found, the solution can quarantine the VM and perform DPI or traffic filtering, avoiding to block the communications to the VM. Security indexes are specified for all VMs depending on factors such as connectivity, number of vulnerabilities, and respective CVSS. VMs can be profiled to obtain detailed information about their state, services running, open ports, and so on. The connection of a VM with other VMs is a crucial aspect that counts toward its profile. Any VM that is connected to a greater number of machines is more critical than one that is connected to fewer VMs since the result of a highly connected VM breach might cause more damage to the system. Knowledge about services operating on a VM is also necessary to validate the alerts related to it. Another important factor is the number of open ports on the machine, as these are highly targeted by cyber-attacks.

In the control plane, we found that although there are already some studies that present the use of risk indicators, this area still requires much more attention for the reasons mentioned above in IV-A. In addition, considering some concern on the part of some authors to measure the impact of their proposals on system performance in terms of latency and throughput, there is enough room to go deeper in that direction as future work.

The work cloud visualized in Fig. 10 highlights the more important terms referred in our analyzed literature. These terms are sized in the figure according to the frequency of their appearance in the surveyed papers. We have excluded from our extensive analysis the more popular non-technical terms used in the English writing (e.g. the, that, etc.). Analyzing Fig. 10, the research community has been much more focused on the mitigation of network security attacks, eventually enhanced by SDN-based solutions, but significantly less involved in the detection of server vulnerabilities, before these could be explored by attackers.

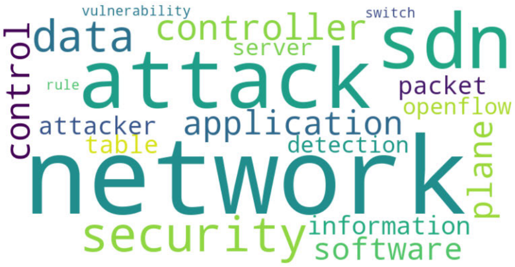

FIGURE 10. Top-20 word cloud from the analyzed literature.

V. OPEN ISSUES

From our literature revision, the most part of the revised work aims to guarantee the system security in a reactive way, except for some work proposing mitigation techniques such as MTD or other similar obfuscation methods to mislead eventual attackers. This means the great majority of the proposed solutions try to do their best in successfully detecting and mitigating running attacks. Thus, there is a strong and generic need to investigate new self-adaptive programmable solutions, offering proactive prevention of cyber menaces before their concretization and without penalizing too much the system normal performance.

In continuation of exploring proactive approaches to enhance cybersecurity, our discussion extends to emerging research areas such as programmable smart contracts integrated with distributed ledger databases. While this innovation promises decentralized and more scalable transaction systems, it faces challenges such as frontrunning attacks due to the reliance on public mining phases for ledger updates. These attacks allow to extract some amount of the victim initial transaction's expected outcome towards directly the attacker(s) profit [94]. Other attacks are discussed in [95]. Recent efforts propose incentivizing good behavior among network participants to preemptively mitigate those attacks. However, to fully realize the potential of these solutions, further investigation is needed. Specifically, there's a need to delve into enhancing smart programmable systems with capabilities for proactive discovery and mitigation of security vulnerabilities at the network edge. This involves leveraging secure smart contracts [96] and trustful mining [96] mechanisms to facilitate the seamless sharing of system assets among stakeholders at the network periphery.

The Digital Twin (DT) can be very useful to elect a set of system management policies to detect and mitigate system vulnerabilities inside the network infrastructure of any organization. The main concept behind DT is representing the real system in a virtual model where future security actions are simulated, tested, selected, and transposed back to the real system. Thus, DT can be fundamentally effective in scenarios associated to high-secure cyber-physical systems [97], using IoT devices, smart agents, and the network edge.

The underneath discussion outlines a roadmap for advancing smart programmable systems towards proactive security measures in next-generation networked systems.

A. DATA PLANE

Leveraging Data Plane programmable technologies, such as the case of P4 [16], [17], can advance the state-of-the-art for the proactive discovery and mitigation of security vulnerabilities in next-generation networks with IoT devices. In fact, the P4 adoption allows the major processing associated to both vulnerability detection and reaction could be done at the data plane level, reducing the necessary system time to react against threats and diminishing the SDN channel control workload. We envision the next P4 data plane open issues:

P4-Based Intrusion Detection Systems (IDS): Explore the development of P4-based IDS [98] tailored for next-generation networks with IoT devices. These systems should be capable of analyzing network traffic in real-time, identifying potential security threats, and triggering proactive mitigation actions.

Dynamic Security Policy Enforcement: Investigate methods to dynamically adapt security policies in P4-enabled network devices based on detected vulnerabilities and network conditions. This involves designing flexible P4 programs that can be updated on-the-fly to address emerging security risks.

Vulnerability-Aware Routing and Traffic Engineering: Develop P4-based mechanisms for integrating vulnerability information into routing and traffic engineering decisions. This could involve leveraging vulnerability data to optimize traffic flows, minimize attack surface, and enhance network resilience against security threats.

Fine-Grained Access Control and Segmentation: Explore the use of P4 to implement fine-grained access control and segmentation strategies in next-generation networks. This includes defining P4-based policies to isolate IoT devices, enforce least privilege principles, and prevent lateral movement of attackers within the network.

Behavioral Anomaly Detection: Investigate the use of P4 for implementing behavioral anomaly detection techniques tailored for IoT environments. Develop P4 programs capable of profiling normal device behavior, detecting deviations indicative of security breaches, and triggering proactive responses.

Integration With Machine Learning: Explore synergies between P4-based network programmability and machine learning techniques for security. Investigate approaches to integrate ML models into P4 pipelines for enhancing the accuracy of security threat detection and mitigation.

Resilience and Fail-Safe Mechanisms: Design P4-based resilience mechanisms to ensure the reliability and fail-safe operation of security features in next-generation networks. This involves implementing redundancy, failover, and recovery mechanisms within P4 programs to withstand attacks and hardware failures.

Scalability and Performance Optimization: Investigate techniques to optimize the scalability and performance of P4-based security solutions in large-scale IoT deployments. This includes designing efficient and predictable packet processing pipelines [99].

Using DPI in P4-Enabled SDN Systems: Enable innovative solutions for dealing with scenarios involving hosts security vulnerabilities and cyber threats trying to explore those. One possible application of DPI's capabilities is that after detecting that a machine has low-risk vulnerabilities, instead of quarantining the machine, by moving it to a VLAN with restricted access, it can be protected by inspecting and eventually discarding malicious traffic destined to that machine. In this way, the machine remains attached in the normal way to the network and it continues available to perform its tasks, until the minor machine vulnerabilities could be solved.

B. CONTROL

Considering now using the upper layers of programmable systems with the intent of advancing the state-of-the-art for the proactive discovery and mitigation of security vulnerabilities in next-generation networks with IoT devices, we envision the next open issues:

Dynamic Security Policy Enforcement: Develop mechanisms within SDN controllers to dynamically enforce security policies based on real-time threat intelligence and network conditions. This involves creating flexible policy management frameworks that can adapt to evolving security threats.

Vulnerability-Aware Network Management: Investigate methods to integrate vulnerability assessment data into SDN controllers for proactive network management. Develop algorithms to prioritize and mitigate vulnerabilities in IoT devices based on their criticality and impact on network security.

Adaptive Access Control: Explore the use of SDN controllers to implement adaptive access control mechanisms for IoT devices. Develop policies that dynamically adjust device access privileges based on contextual information, such as device behavior and security goals.

Behavioral Anomaly Detection: Research techniques for implementing behavioral anomaly detection within SDN controllers. Develop algorithms to analyze network traffic patterns, detect abnormal behavior indicative of security threats, and trigger appropriate mitigation actions.

Threat Intelligence Integration: Investigate how to enhance the SDN controllers' intelligence for dealing with system menaces. A possible way is developing novel mechanisms to automatically update security policies based on emerging threats and known vulnerabilities in IoT devices. Easily to configure and Unified threat management systems are seen as very important to keep protected the operation of networked systems [100].

Software-Defined Segmentation: Explore the use of SDN controllers to implement software-defined segmentation in next-generation networks, using segmentation technologies such as VLAN, segment routing or network slicing [101], [102], [103]. Develop policies to isolate IoT devices into secure segments based on their security requirements and communication patterns.

Machine Learning Integration: Investigate synergies between SDN controllers and machine learning techniques for security enhancement [104]. Develop algorithms to analyze network data, detect anomalies, and predict potential security threats, leveraging the programmability of SDN controllers.

Resilience Mechanisms: Design resilience mechanisms within SDN controllers to ensure the reliability of system security features. Develop robust recovery mechanisms to system threats such as cyber attacks and system failures [105].

Scalability and Performance Optimization: Optimize the scalability and performance of SDN controllers in large-scale IoT deployments. Develop algorithms and data structures to efficiently handle the increased volume of security-related data and policy updates.

Validation and Testing Frameworks: Develop comprehensive validation and testing frameworks for SDN-based security solutions, including the P4 usage at the data plane. This involves creating realistic testbeds, generating attack scenarios, and evaluating the effectiveness of proactive security measures. Then, the more efficient solutions can be applied in the production network.

As a final conclusion of this section, to tackle the long list of unresolved problems discussed above, we think a strategic cooperation is also necessary among scholars, industry participants, and policymakers to foster innovation, develop best practices, and establish standards for proactively addressing key security vulnerabilities in edge computing scenarios before these system weaknesses be explored by cyber attackers.

VI. CONCLUSION

The current manuscript has comprehensively revised the available literature for both detection and mitigation of system vulnerabilities, and their associated risks. From the literature analysis, we identified, classified, discussed, and compared the more prominent proposals.

Resulting from the analysis made on the surveyed work, we have identified a large list of future interesting research directions. These are following summarized. There is a strong need on incorporating active probing tools into SDN to quickly identify vulnerabilities, prioritize efforts to mitigate them and adopt proactive instead reactive countermeasures. The lack of risk indicators in data plane mitigation reveals the need for a more sophisticated method of evaluating and addressing security concerns. Utilizing the functionalities of the data plane not only resolves problems related to latency but also enhances the capacity to handle larger workloads and adjust to changing scenarios, building a solid foundation for an effective security framework. In addition, considering the control plane, although several studies mention the utilization of risk indicators, the existing literature lacks complete solutions that demonstrate measurable effects on network performance. Further developments are also needed in programmable systems supported by decision consensus among distributed agents for better protecting the system and its sensitive data against sophisticated security threats at the network periphery.

REFERENCES

[1] A. Fleck. *Infographic: Cybercrime Expected to Skyrocket in Coming Years*. Statista Daily Data. Accessed: Jan. 2, 2024. [Online]. Available: https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/

[2] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

[3] *Perform Systematic Literature Reviews*. Parsifal. Accessed: Jan. 2, 2024. [Online]. Available: https://parsif.al/

[4] H. Sabino, R. V. S. Almeida, L. B. D. Moraes, W. P. D. Silva, R. Guerra, C. Malcher, D. Passos, and F. G. O. Passos, "A systematic literature review on the main factors for public acceptance of drones," *Technol. Soc.*, vol. 71, Nov. 2022, Art. no. 102097.

[5] M. H. Kashani and E. Mahdipour, "Load balancing algorithms in fog computing," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1505–1521, Mar. 2023.

[6] M. Kaur and R. Aron, "A systematic study of load balancing approaches in the fog computing environment," *J. Supercomput.*, vol. 77, no. 8, pp. 9202–9247, Aug. 2021.

[7] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: Threats, mitigations, and future directions," *J. Reliable Intell. Environ.*, vol. 9, no. 2, pp. 201–239, Jun. 2023.

[8] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1744–1772, 2nd Quart., 2019.

[9] N. Sun, M. Ding, J. Jiang, W. Xu, X. Mo, Y. Tai, and J. Zhang, "Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1748–1774, 3rd Quart., 2023.

[10] M. Ghaznavi, E. Jalalpour, M. A. Salahuddin, R. Boutaba, D. Migault, and S. Preda, "Content delivery network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2166–2190, 4th Quart., 2021.

[11] E. Bardhi, M. Conti, R. Lazzeretti, and E. Losiouk, "Security and privacy of IP-ICN coexistence: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2427–2455, 4th Quart., 2023.

[12] W. Li, W. Meng, and L. F. Kwok, "A survey on OpenFlow-based software defined networks: Security challenges and countermeasures," *J. Netw. Comput. Appl.*, vol. 68, pp. 126–139, Jun. 2016.

[13] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, 4th Quart., 2015.

[14] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," *J. Netw. Comput. Appl.*, vol. 67, pp. 1–25, May 2016.

[15] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[16] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. E. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," 2013, *arXiv:1312.1719*.

[17] A. Liatifis, P. Sarigiannidis, V. Argyriou, and T. Lagkas, "Advancing SDN from OpenFlow to p4: A survey," *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–37, Sep. 2023.

[18] *Cvss V3.1 Specification Document*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.first.org/cvss/v3.1/specification-document

[19] *CVE*. Accessed: Feb. 10, 2024. [Online]. Available: https://www.cve.org/

[20] V. Pham and T. Dang, "CVExplorer: Multidimensional visualization for common vulnerabilities and exposures," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1296–1301.

[21] G. Bartlett, J. Heidemann, and C. Papadopoulos, "Understanding passive and active service discovery," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, Oct. 2007, pp. 57–70.

[22] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Attack graph-based moving target defense in software-defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1653–1668, Sep. 2020.

[23] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1909–1941, 3rd Quart., 2020.

[24] A. Chowdhary, S. Pisharody, and D. Huang, "SDN based scalable MTD solution in cloud network," in *Proc. ACM Workshop Moving Target Defense*, Oct. 2016, pp. 27–36.

[25] L. Spitzner, "The honeynet project: Trapping the hackers," *IEEE Secur. Privacy*, vol. 1, no. 2, pp. 15–23, Mar. 2003.

[26] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3906–3919, Dec. 2018.

[27] R. T. El-Maghraby, N. M. Abd Elazim, and A. M. Bahaa-Eldin, "A survey on deep packet inspection," in *Proc. 12th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2017, pp. 188–197.

[28] P. Krongbaramee and Y. Somchit, "Implementation of SDN stateful firewall on data plane using open vSwitch," in *Proc. 15th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2018, pp. 1–5.

[29] P. Mehran, E. A. Reza, and B. Laleh, "SPKT: Secure port knock-tunneling, an enhanced port security authentication mechanism," in *Proc. IEEE Symp. Comput. Informat. (ISCI)*, Mar. 2012, pp. 145–149.

[30] Z. Niu, Q. Li, C. Ma, H. Li, H. Shan, and F. Yang, "Identification of critical nodes for enhanced network defense in MANET-IoT networks," *IEEE Access*, vol. 8, pp. 183571–183582, 2020.

[31] A. N. Alhaj and N. Dutta, "Analysis of security attacks in SDN network: A comprehensive survey," in *Contemporary Issues in Communication, Cloud and Big Data Analytics* (Lecture Notes in Networks and Systems), vol. 281. Singapore: Springer, 2022, pp. 27–37.

[32] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[33] S. Dong, K. Abbas, and R. Jain, "A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments," *IEEE Access*, vol. 7, pp. 80813–80828, 2019.

[34] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[35] J. C. C. Chica, J. C. Imbachi, and J. F. B. Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, Jun. 2020, Art. no. 102595.

[36] I. Alsmadi and D. Xu, "Security of software defined networks: A survey," *Comput. Secur.*, vol. 53, pp. 79–108, Sep. 2015.

[37] N. M. Yungaicela-Naula, C. Vargas-Rosales, J. A. Pérez-Díaz, and M. Zareei, "Towards security automation in software defined networks," *Comput. Commun.*, vol. 183, pp. 64–82, Feb. 2022.

[38] M. Yu, J. Zhuge, M. Cao, Z. Shi, and L. Jiang, "A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices," *Future Internet*, vol. 12, no. 2, p. 27, Feb. 2020.

[39] R. M. Ogunnaike and B. Lagesse, "Toward consumer-friendly security in smart environments," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 612–617.

[40] T. Eom, J. B. Hong, S. An, J. S. Park, and D. S. Kim, "A systematic approach to threat modeling and security analysis for software defined networking," *IEEE Access*, vol. 7, pp. 137432–137445, 2019.

[41] Y. Nikoloudakis, E. Pallis, G. Mastorakis, C. X. Mavromoustakis, C. Skianis, and E. K. Markakis, "Vulnerability assessment as a service for fog-centric ICT ecosystems: A healthcare use case," *Peer Peer Netw. Appl.*, vol. 12, no. 5, pp. 1216–1224, Sep. 2019.

[42] S. Kim, S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "DIVERGENCE: Deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4834–4846, Dec. 2022.

[43] R. Sood and S. S. Kang, "Detection and mitigation of network vulnerability through scheduling framework in software defined networks," in *Proc. 6th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2019, pp. 1118–1123.

[44] M. Zolotukhin, T. Hämäläinen, and R. Immonen, "Curious SDN for network attack mitigation," in *Proc. IEEE 21st Int. Conf. Softw. Quality, Rel. Secur. Companion (QRS-C)*, Dec. 2021, pp. 630–635.

[45] D. T. Thu Hien, H. Do Hoang, and V.-H. Pham, "Empirical study on reconnaissance attacks in SDN-aware network for evaluating cyber deception," in *Proc. RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Aug. 2021, pp. 1–6.

[46] L. M. Almutairi and S. Shetty, "Generalized stochastic Petri net model based security risk assessment of software defined networks," in *Proc. MILCOM IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 545–550.

[47] Q. Qin, K. Poularakis, and L. Tassiulas, "A learning approach with programmable data plane towards IoT security," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 410–420.

[48] S.-Y. Chang, Y. Park, and A. Muralidharan, "Fast address hopping at the switches: Securing access for packet forwarding in SDN," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 454–460.

[49] S.-Y. Chang, Y. Park, and B. B. A. Babu, "Fast IP hopping randomization to secure hop-by-hop access in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 308–320, Mar. 2019.

[50] E. O. Zaballa, D. Franco, Z. Zhou, and M. S. Berger, "P4Knocking: Offloading host-based firewall functionalities to the network," in *Proc. 23rd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2020, pp. 7–12.

[51] I. Pali and R. Amin, ''PortSec: Securing port knocking system using sequence mechanism in SDN environment,'' in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2022, pp. 1009–1014.

[52] A. Saxena, R. Muttreja, S. Upadhyay, K. S. Kumar, and U. Venkanna, ''P4Filter: A two level defensive mechanism against attacks in SDN using P4,'' in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2021, pp. 113–118.

[53] J. Li, H. Jiang, W. Jiang, J. Wu, and W. Du, ''SDN-based stateful firewall for cloud,'' in *Proc. IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2020, pp. 157–161.

[54] A. Javadpour, F. Ja'fari, T. Taleb, and M. Shojafar, ''A cost-effective MTD approach for DDoS attacks in software-defined networks,'' in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 4173–4178.

[55] H. Galadima, A. Seeam, and V. Ramsurrun, ''Cyber deception against DDoS attack using moving target defence framework in SDN IoT-EDGE networks,'' in *Proc. 3rd Int. Conf. Next Gener. Comput. Appl. (NextComp)*, Oct. 2022, pp. 1–6.

[56] K. Guo, Y. Gao, D. Wang, H. Zhi, T. Zhang, and Y. Lu, ''A hybrid routing mutation mechanism based on mutation cost and resource trustworthiness in network moving target defense,'' in *Proc. 7th IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, Jul. 2022, pp. 426–431.

[57] T. Moghaddam, M. Kim, J.-H. Cho, H. Lim, T. J. Moore, F. F. Nelson, and D. D. Kim, ''A practical security evaluation of a moving target defence against multi-phase cyberattacks,'' in *Proc. 52nd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2022, pp. 103–110.

[58] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Nelson, and H. Lim, ''Poster: Address shuffling based moving target defense for in-vehicle software-defined networks,'' in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2019, pp. 1–3.

[59] M. F. Hyder and M. A. Ismail, ''Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches,'' *IEEE Access*, vol. 9, pp. 21881–21894, 2021.

[60] F. S. D. Silva, T. Pascoal, E. P. Neto, R. S. S. Nunes, C. H. M. Souza, and A. Neto, ''An adaptive moving target defense approach for software-defined networking protection,'' in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, May 2023, pp. 1–6.

[61] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J.-H. Cho, and T. J. Moore, ''FRVM: Flexible random virtual IP multiplexing in software-defined networks,'' in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 579–587.

[62] J. Narantuya, S. Yoon, H. Lim, J.-H. Cho, D. S. Kim, T. Moore, and F. Nelson, ''SDN-based IP shuffling moving target defense with multiple SDN controllers,'' in *Proc. 49th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Supplemental Volume (DSN-S)*, Jun. 2019, pp. 15–16.

[63] S. Chiba, L. Guillen, S. Izumi, T. Abe, and T. Suganuma, ''Design of a network scan defense method by combining an SDN-based MTD and IPS,'' in *Proc. 22nd Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2021, pp. 273–278.

[64] B. Zhang, L. Han, and S. Sun, ''Dynamic random route mutation mechanism for moving target defense in SDN,'' in *Proc. 6th Int. Symp. Comput. Inf. Process. Technol. (ISCIPT)*, Jun. 2021, pp. 536–541.

[65] M. Kim, J.-H. Cho, H. Lim, T. J. Moore, F. F. Nelson, R. K. L. Ko, and D. Dongseong Kim, ''Evaluating performance and security of a hybrid moving target defense in SDN environments,'' in *Proc. IEEE 22nd Int. Conf. Softw. Quality, Rel. Secur. (QRS)*, Dec. 2022, pp. 276–286.

[66] S. Debroy, P. Calyam, M. Nguyen, A. Stage, and V. Georgiev, ''Frequency-minimal moving target defense using software-defined networking,'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–6.

[67] C. Gudla and A. H. Sung, ''Moving target defense application and analysis in software-defined networking,'' in *Proc. 11th IEEE Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2020, pp. 0641–0646.

[68] C. Gudla and A. H. Sung, ''Moving target defense discrete host address mutation and analysis in SDN,'' in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2020, pp. 55–61.

[69] S. Macwan and C.-H. Lung, ''Investigation of moving target defense technique to prevent poisoning attacks in SDN,'' in *Proc. IEEE World Congr. Services (SERVICES)*, Jul. 2019, pp. 178–183.

[70] J. B. Hong, S. Yoon, H. Lim, and D. S. Kim, ''Optimal network reconfiguration for software defined networks using shuffle-based online MTD,'' in *Proc. IEEE 36th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2017, pp. 234–243.

[71] T. Moghaddam, G. Yang, C. Thapa, S. Camtepe, and D. D. Kim, ''POSTER: Toward intelligent cyber attacks for moving target defense techniques in software-defined networking,'' in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Jul. 2023, pp. 1022–1024.

[72] D. P. Sharma, J.-H. Cho, T. J. Moore, F. F. Nelson, H. Lim, and D. S. Kim, ''Random host and service multiplexing for moving target defense in software-defined networks,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[73] M. M. Islam, Q. Duan, and E. Al-Shaer, ''Specification-driven moving target defense synthesis,'' in *Proc. 6th ACM Workshop Moving Target Defense*, Nov. 2019, pp. 13–24.

[74] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, ''MTD analysis and evaluation framework in software defined network (MASON),'' in *Proc. ACM Int. Workshop Secur. Softw. Defined Netw. Netw. Function Virtualization*, Mar. 2018, pp. 43–48.

[75] H. Kim, E. Hwang, D. Kim, J.-H. Cho, T. J. Moore, F. F. Nelson, and H. Lim, ''Time-based moving target defense using Bayesian attack graph analysis,'' *IEEE Access*, vol. 11, pp. 40511–40524, 2023.

[76] M. Ge, J.-H. Cho, D. Kim, G. Dixit, and I.-R. Chen, ''Proactive defense for Internet-of-Things: Moving target defense with cyberdeception,'' *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–31, Feb. 2022.

[77] Z. Rehman, I. Gondal, M. Ge, H. Dong, M. Gregory, and Z. Tari, ''Proactive defense mechanism: Enhancing IoT security through diversity-based moving target defense and cyber deception,'' *Comput. Secur.*, vol. 139, Apr. 2024, Art. no. 103685.

[78] F. Shi, Z. Zhou, W. Yang, S. Li, Q. Liu, and X. Bao, ''AHIP: An adaptive IP hopping method for moving target defense to thwart network attacks,'' in *Proc. 26th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2023, pp. 1300–1305.

[79] L. Zhang, Q. Wei, K. Gu, and H. Yuwen, ''Path hopping based SDN network defense technology,'' in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 2058–2063.

[80] J. H. Anajemba, N. Ababneh, Y. Hamid, A. Chowhan, O. Obinna, and E. Vajzovic, ''SDN-based port hopping technique for mitigating network attacks,'' in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2023, pp. 1–6.

[81] Z. Minjiao, M. Yufeng, W. Bo, and Q. Zhang, ''A dynamic deceptive honeynet system with a hybrid of virtual and real devices,'' in *Proc. 5th Int. Conf. Comput. Big Data (ICCBD)*, Dec. 2022, pp. 113–117.

[82] W. Huang, Y. Sun, W. Ou, and Y. Wang, ''A flow scheduling model for SDN honeypot using multi-layer attack graphs and signaling game,'' in *Proc. 7th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2021, pp. 2012–2020.

[83] B. Park, S. P. Dang, S. Noh, J. Yi, and M. Park, ''Dynamic virtual network honeypot,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 375–377.

[84] S. Kyung, W. Han, N. Tiwari, V. H. Dixit, L. Srinivas, Z. Zhao, A. Doupé, and G.-J. Ahn, ''HoneyProxy: Design and implementation of next-generation honeynet via SDN,'' in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 1–9.

[85] X. Luo, Q. Yan, M. Wang, and W. Huang, ''Using MTD and SDN-based honeypots to defend DDoS attacks in IoT,'' in *Proc. Comput., Commun. IoT Appl. (ComComAp)*, Oct. 2019, pp. 392–395.

[86] C. Gao, Y. Wang, X. Xiong, and W. Zhao, ''MTDCD: An MTD enhanced cyber deception defense system,'' in *Proc. IEEE 4th Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, vol. 4, Jun. 2021, pp. 1412–1417.

[87] B. Rashidi, C. Fung, K. W. Hamlen, and A. Kamisinski, ''HoneyV: A virtualized honeynet system based on network softwarization,'' in *Proc. NOMS - IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–5.

[88] C. S. Bontaş, I.-M. Stan, and R. Rughiniş, ''Honeypot generator using software defined networks and recursively defined topologies,'' in *Proc. 21st RoEduNet Conf., Netw. Educ. Res. (RoEduNet)*, Sep. 2022, pp. 1–5.

[89] J. Franco, A. Aris, L. Babun, and A. S. Uluagac, ''S-pot: A smart honeypot framework with dynamic rule configuration for SDN,'' in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 2818–2824.

[90] R. Li, M. Zheng, D. Bai, and Z. Chen, ''SDN based intelligent honeynet network model design and verification,'' in *Proc. Int. Conf. Mach. Learn. Intell. Syst. Eng. (MLISE)*, Jul. 2021, pp. 59–64.

[91] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 1602–1606.

[92] M. Karakate, H. Esaki, and H. Ochiai, "SDNHive: A proof-of-concept SDN and honeypot system for defending against internal threats," in *Proc. 11th Int. Conf. Commun. Netw. Secur.*, Dec. 2021, pp. 9–20.

[93] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Depend. Secure Comput.*, vol. 10, no. 4, pp. 198–211, Jul. 2013.

[94] C. F. Torres and R. Camino, "Frontrunner Jones and the raiders of the dark forest: An empirical study of frontrunning on the Ethereum blockchain," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 1343–1359.

[95] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 198–214.

[96] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50759–50779, 2019.

[97] M. Eckhart and A. Ekelhart, "Digital twins for cyber-physical systems security: State of the art and outlook," in *Security and Quality in Cyber-Physical Systems Engineering*. Cham, Switzerland: Springer, 2019, pp. 383–412.

[98] B. Lewis, M. Broadbent, and N. Race, "P4ID: P4 enhanced intrusion detection," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2019, pp. 1–4.

[99] H. Harkous, M. Jarschel, M. He, R. Pries, and W. Kellerer, "p8: P4 with predictable packet processing performance," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 2846–2859, Sep. 2021.

[100] A. Siddiqui, B. P. Rimal, M. Reisslein, and Y. Wang, "Survey on unified threat management (UTM) systems for home networks," *IEEE Commun. Surveys Tuts.*, early access, Mar. 28, 2024, doi: 10.1109/COMST.2024.3382470.

[101] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[102] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, May 2017.

[103] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[104] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[105] A. S. da Silva, P. Smith, A. Mauthe, and A. Schaeffer-Filho, "Resilience support in software-defined networking: A survey," *Comput. Netw.*, vol. 92, pp. 189–207, Dec. 2015.

JOÃO POLÓNIO received the B.Sc. degree in telecommunications and computer engineering from the ISCTE—Instituto Universitario de Lisboa, Portugal, in 2021, where he is currently pursuing the M.Sc. degree in telecommunications and computer engineering. During his undergraduate studies, he demonstrated a keen interest in the field of computer networks and network security. Since 2022, he has actively been assisting in teaching computer network architectures.

JOSÉ MOURA received the B.Sc. degree in electronics and telecommunications from Universidade de Aveiro, Portugal, in 1989, the M.Sc. degree in computer networks from the Faculdade de Engenharia, Universidade do Porto, Portugal, in 2001, and the Ph.D. degree in computer science from Lancaster University, U.K., in 2011. From 1989 to 2000, he was an Engineer in supervisory control and data acquisition (SCADA) systems and industrial automation with EFACEC Sistemas Electronica, Portugal. From 2000 to 2001, he was a Researcher with INESC, Porto, Portugal. Since 2001, he has been teaching computer networks with the ISCTE—Instituto Universitario Lisboa, Portugal, and a Researcher with the Instituto de Telecomunicações, Portugal. His current research interests include network management, edge computing, optimization, virtualization, software-defined networking, and resilience on networked systems. He is an active reviewer for several Quartile 1 journals.

RUI NETO MARINHEIRO received the M.Eng. degree in electrical and computer science engineering (telecommunications) from the Faculty of Engineering, University of Porto, Portugal, and the Ph.D. degree in multimedia information systems from the University of Southampton, U.K. He is currently an Associate Professor with the ISCTE—Lisbon University Institute, Portugal, and a Researcher with the Instituto de Telecomunicações. He has coordinated and participated in many national and international research projects. He has extensive experience in teaching and researching in the fields of telecommunications, computer networks, and hypermedia.