DDoS Mitigation in Kubernetes: A Review of Extended Berkeley Packet Filtering and eXpress Data Path Technologies

Mircea Ţălu [1,*)]

[1)] Faculty of Automation and Computer Science, The Technical University of Cluj-Napoca, Cluj County, Romania

[1)] SC ACCESA IT SYSTEMS SRL, Cluj-Napoca, Romania

E-mail: talu.s.mircea@gmail.com[1)], mircea.talu@accesa.eu[1)]

ABSTRACT

Kubernetes, a widely adopted platform for container orchestration, faces increasing threats from sophisticated cyberattacks, such as Distributed Denial of Service (DDoS) attacks, which can significantly impact the stability, availability, and operational integrity of clusters. These attacks may overwhelm the cluster's control plane, disrupt pod scheduling, or exhaust network resources. To address these challenges, emerging Linux kernel technologies like the Extended Berkeley Packet Filter (eBPF) and eXpress Data Path (XDP) offer promising solutions by enabling high-performance packet filtering, real-time traffic analysis, and advanced intrusion detection within the kernel. These technologies reduce latency, improve resource efficiency, and strengthen the overall security of cloud-native environments. This review explores the integration of eBPF and XDP for DDoS mitigation in Kubernetes, analyzing current research, identifying limitations, and highlighting their potential to establish scalable, adaptive, and efficient mitigation frameworks. By incorporating these insights, the development of robust, tailored security policies for modern containerized infrastructures can be better informed and implemented.

Keywords: Distributed Denial of Service (DDoS) attacks, eBPF (Extended Berkeley Packet Filter), kubernetes clusters, linux kernel technologies, XDP (eXpress Data Path)

1. Introduction

Before the 1990s, network packet processing faced critical limitations due to reliance on classic packet filtering mechanisms. These approaches required transferring every packet from the kernel space to the userspace for inspection, resulting in significant computational overhead, limited scalability, and inefficiency in high-throughput environments [1], [2].

Addressing these challenges, Steven McCanne and Van Jacobson developed the Berkeley Packet Filter (BPF) in 1992–1993. This innovation introduced an in-kernel virtual machine (VM) capable of executing user-defined packet filtering programs directly within the kernel. By processing packets at this level, BPF significantly reduced data copying, improved performance, and established a foundation for scalable and efficient network monitoring tools.

This architecture not only minimized data copying but also enhanced performance by processing only the relevant packets at the kernel level [3], [4]. As a result, BPF set the stage for more scalable and efficient network monitoring and diagnostic tools, forming the foundational technology for modern advancements like eBPF, which extends these principles to more complex use cases, including security, performance profiling, and traffic management.

The introduction of the Extended Berkeley Packet Filter (eBPF) in 2013, spearheaded by Alexei Starovoitov, marked a transformative evolution of the original BPF [5]. eBPF enhanced the classic BPF's (cBPF) capabilities by enabling dynamic code injection into the Linux kernel at runtime, allowing the system to respond to specific

* Corresponding author.
Received: March 24th, 2025. Revised: April 14th, 2025. Accepted: May 9th, 2025.

Available online: July 8th, 2025.

DOI: https://doi.org/10.12962/j24068535.v23i1.a1268

kernel events such as network traffic, system calls, and hardware interrupts. Unlike cBPF, which was limited to basic packet filtering tasks, eBPF extended functionality to support complex operations, broadening its applicability to areas such as security, performance profiling, and traffic management Integrated into the Linux kernel with version 3.18, eBPF marks a significant milestone in system observability and optimization, introducing a modern instruction set architecture (ISA) with enhanced features, including additional registers, to boost efficiency, flexibility, and functionality. It reduces the reliance on resource-intensive context switching between kernel and userspace, thereby minimizing latency and enhancing overall efficiency. Programs written for eBPF are developed in a constrained subset of the C programming language, compiled into bytecode, and executed securely within the kernel's sandboxed environment. Also, eBPF offers predefined data structures, like hash maps, LRU maps, and arrays, that can be accessed and modified by both kernel and userspace programs, allowing dynamic adaptation to changing system conditions. This architecture ensures safety while enabling advanced monitoring, filtering, and diagnostic operations without requiring kernel modifications [6]. However, the BPF Compiler Collection (BCC) simplifies the development of eBPF programs by allowing developers to write these programs using Python along with a restricted version of C, streamlining the process and making eBPF more accessible for various applications [7], [8].

The eXpress Data Path (XDP) is an advanced Linux kernel framework optimized for high-performance packet processing directly at the kernel level. XDP enables the efficient handling of network traffic by intercepting packets as they arrive at the network interface card (NIC), allowing for real-time processing before the data reaches the kernel’s networking stack. This early interception reduces overhead and significantly improves the speed and scalability of packet handling. By bypassing traditional networking layers and leveraging just-in-time (JIT) compilation of eBPF programs, XDP achieves ultra-low latency while preserving the kernel’s stability and security. This makes it an invaluable tool for applications that require precise and efficient network traffic control, such as cloud infrastructures, data centers, and edge computing systems [9].

Kubernetes, a widely adopted open-source orchestration platform, streamlines the management of containerized applications. It provides dynamic scalability and resource allocation for both physical and virtual servers, making it an essential tool for modern infrastructure. Within a Kubernetes cluster, communication is facilitated between containers and the applications they host, ensuring seamless operations. The network architecture is divided into critical components, including the cluster network and the pod network, which collectively support the flow of data across the system. However, its complexity introduces inherent security vulnerabilities that could be exploited if not properly mitigated. For instance, an inadequately secured cluster network may expose the system to malicious traffic injection or unauthorized data interception. Additionally, Kubernetes relies on containers built from images, and any compromise of these images through malicious code could jeopardize the integrity of applications. The Kubelet, responsible for node-to-container communication, represents another critical point that could be exploited in an attack. A poorly secured cluster network can expose sensitive data to attackers or enable malicious traffic injection. Despite these risks, Kubernetes excels in ensuring high availability and scalability, leveraging its microservices-based architecture. Furthermore, it enables developers to evaluate application deployment by dynamically rerouting traffic between containers, enhancing testing and fault tolerance [10], [11]. Kubernetes faces significant security challenges, especially concerning reactive measures that detect policy violations post-occurrence [12]. Existing solutions like Sysdig [13], Falco [14], and KubAnomaly [15] provide reactive or anomaly detection capabilities, while KubeArmor [16] integrates eBPF to enforce runtime security at the container level. These advancements in runtime enforcement systems highlight the potential of eBPF in addressing Kubernetes security vulnerabilities. Furthermore, Open Policy Agent (OPA) functions as a versatile security policy engine [17], while Gatekeeper [18], serving as its Kubernetes-native sidecar, enforces these policies by integrating seamlessly into Kubernetes clusters to ensure compliance and strengthen security measures.

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks have emerged as some of the most prominent and damaging cyber threats in modern computing. These attacks aim to overwhelm servers with malicious traffic, rendering them inaccessible to legitimate users and causing substantial disruption to services.

Beyond creating operational downtime, DDoS attacks impose significant performance overheads, degrading the efficiency and reliability of affected systems [19], [20], [21].

Over the past decade, these attacks have been recognized as a major security risk, capable of not only denying access to network resources but also leading to the complete failure of the targeted networks. Cloud computing environments and microservices architectures are particularly susceptible to DDoS attacks due to their reliance on distributed systems and open connectivity [22], [23], [24]. Attackers exploit unprotected or under-secured entry points, leveraging these vulnerabilities to execute large-scale attacks that compromise entire systems. Notably, these cyber threats are not constrained to specific sectors [25], [26], [27]; they transcend boundaries and adapt to various environments, increasing their scope and impact [28], [29].

Research has highlighted that the global and multi-faceted nature of DDoS attacks enables attackers to exploit less-privileged loopholes in systems, compromising them at an alarming scale. As a result, addressing the vulnerabilities exploited by DDoS attacks has become a critical focus for modern network security strategies [19], [21], [23]. In response, technologies like eBPF (Extended Berkeley Packet Filter) have revolutionized system performance and network security by enabling packet-level filtering without modifying the kernel [30],[31]. Additionally, the Express Data Path (XDP) enhances security by providing rapid packet processing and programmable pathways for network traffic [32]. As the complexity of cyber threats grows, so does the need for robust solutions to protect Kubernetes clusters and the applications they support. Conventional methods for mitigating DDoS attacks often introduce substantial performance overhead to Kubernetes clusters. This challenge highlights the importance of developing more efficient solutions that can monitor and detect DDoS traffic effectively, ensuring system resilience without compromising cluster performance [33], [34].

The primary research questions guiding this study are:

a) How can eBPF and XDP be effectively integrated into Kubernetes environments for DDoS mitigation?

b) What are the performance implications of using eBPF and XDP for kernel-level packet filtering and intrusion detection in cloud-native architectures?

c) What are the potential challenges and limitations in applying these technologies to large-scale, dynamic Kubernetes clusters?

While numerous studies have explored the general applications of eBPF and XDP technologies within the broader context of network security, there remains a significant gap in the literature concerning their specific integration into Kubernetes environments, particularly for DDoS mitigation. This review addresses this critical gap by presenting a focused, in-depth analysis of how eBPF and XDP are being leveraged to enhance security within containerized, cloud-native infrastructures. Unlike previous surveys, which often treat eBPF/XDP in isolation or provide only a high-level overview, this review offers a systematic and comparative evaluation of their performance in Kubernetes, incorporating key metrics such as throughput, latency, resource utilization, detection accuracy, and false positive rate. Furthermore, the review introduces a comprehensive classification of emerging research directions, including adaptive packet filtering, dynamic observability, microservices-specific anomaly detection, realistic DDoS dataset creation, edge computing optimization, AI-driven policy enforcement, energy-efficient algorithm design, and multi-cloud security integration. These proposed directions are not only grounded in current technological limitations but also informed by recent advancements in the scientific literature, thereby providing a roadmap for future innovation. By synthesizing findings from 36 primary studies and aligning them with practical deployment challenges, the review offers both theoretical insights and applied guidance for researchers and practitioners. As such, this work advances the state of knowledge by establishing a direct correlation between programmable kernel-level technologies and the evolving security needs of Kubernetes-based systems.

2. Research Method

A systematic review was conducted to explore advancements in DDoS mitigation strategies for Kubernetes environments using eBPF and XDP, following five key steps:

a) Formulating research questions;

b) Identifying and collecting relevant literature;

c) Evaluating the quality of the selected studies;

d) Synthesizing the gathered evidence; and

e) Analyzing the findings to derive insights.

To address the increasing relevance of this topic, the review covers literature published from 2017 to 2025, focusing on journal articles and studies that delve into the integration of eBPF and XDP for enhancing network security in Kubernetes systems.

A thorough and methodical literature search was carried out across a selection of esteemed academic databases, including ScienceDirect, SpringerLink, IEEE Xplore, MDPI, Google Scholar, and the ACM Digital Library. This approach was designed to ensure the inclusion of a broad and representative array of studies relevant to the subject of eBPF and XDP applications in Kubernetes security. We formulated a search query string using Boolean operators, incorporating keywords a combination of key terms and phrases such as 'eBPF (Extended Berkeley Packet Filter)', 'XDP (Express Data Path)', 'DDoS Mitigation', 'Kubernetes Security', 'Network Packet Filtering', 'Intrusion Detection and Prevention', 'Cloud-Native Security', 'High-Performance Packet Processing', 'Kubernetes in centralized cloud environments', 'Kernel-Level Security', 'Traffic Anomaly Detection', 'Microservices Observability' to capture a wide scope of pertinent scholarly work.

A total of 48 research papers were initially retrieved. After removing duplicates and irrelevant studies, to maintain the focus and relevance of the review, inclusion criteria were applied, specifying that only the following types of studies were considered: (1) peer-reviewed journal articles or conference papers, (2) works addressing the integration and application of eBPF and XDP technologies in Kubernetes or other cloud-native infrastructures, (3) studies offering empirical data, case studies, or experimental results that directly relate to the research questions, and (4) publications within the past decade, ensuring the consideration of the most current advancements in the field.

Conversely, studies were excluded based on any of the following conditions: (1) non-English language publications, (2) works that did not specifically address the security challenges and solutions within Kubernetes or cloud-native environments, (3) opinion articles, editorials, or publications that lacked empirical findings, and (4) publications published prior to 2017, thereby preserving the relevance and timeliness of the review in reflecting the latest trends and technological developments. After applying these criteria, 36 papers were selected for an in-depth review. The results of this review provided insights into the effectiveness, performance trade-offs, and integration challenges of eBPF/XDP in Kubernetes environments, forming the foundation for the comparative analysis presented in this review.

2.1. Advancements in eBPF and XDP for Kubernetes Security

Recent studies have emphasized the integration of advanced technologies such as eBPF and XDP to strengthen security in Kubernetes environments. These tools provide cutting-edge solutions for handling complex network operations, detecting intrusions, and mitigating DDoS attacks. Their application extends across multiple facets of modern network security, including packet filtering, intrusion detection, and optimizing data flow management within containerized services, creating a continuous and interconnected framework for enhanced protection and efficiency. Table 1 highlights the main focus, methods, and outcomes of previous studies, providing a clear overview for comparison.

Bertin [35] presented a solution developed by the Cloudflare DDoS mitigation team to handle large-scale DDoS attacks. The approach relies on kernel bypass and classic BPF, enabling packet filtering in userspace while bypassing the standard packet processing mechanisms of Netfilter and the Linux network stack. This method addresses performance bottlenecks encountered when relying solely on traditional Linux kernel features for managing large packet floods.

Table 1: Main focus, methods, and outcomes of previous studies concerning advancements in eBPF and XDP for Kubernetes security

Study/Approach	Key Focus	Technology/Method	Findings/Results
Bertin [35] (Cloudflare)	DDoS Mitigation	Kernel bypass, BPF	Efficient packet filtering in userspace, bypassing Netfilter, addresses performance bottlenecks.
Koksal et al. [36] (MEC Networks)	DDoS Mitigation in MEC Networks	Containerized Network Functions (CNFs), IDPS, Kubernetes	Scalable defense using CNFs and auto-scaling in MEC environments, effective in real-world setups.
Miano et al. [6] (eBPF for Monitoring)	Kernel-Level Monitoring	eBPF	eBPF for flexible data processing, challenges in complex network functions.
Miano et al. [37] (SmartNICs for DDoS)	User-space Packet Filtering	SmartNICs, eBPF, XDP	SmartNICs reduce server load but require additional components for complete DDoS mitigation.
Hohlfeld et al. [38] (Packet Offloading)	Offloading Mechanisms in Linux Kernel	XDP, SmartNICs, AF XDP	Offloading improves small task processing, but excessive load on SmartNICs leads to performance degradation.
Liu et al. [39] (Network Observability)	Microservices Observability	eBPF	Non-intrusive network observability with low system impact, uses machine learning for performance analysis.
Miano et al. [40] (Polycube - NFV)	Network Function Virtualization (NFV)	Polycube, eBPF	Flexible in-kernel NFV, strong isolation and composability for cloud environments.
Wang & Chang [41] (IDS with eBPF)	Intrusion Detection System (IDS)	eBPF, modified Snort ruleset	eBPF-based IDS outperforms traditional Snort in throughput and efficiency.
Budigiri et al. [42] (eBPF-powered Solutions)	Security in Cloud-native Environments	eBPF, Calico, Cilium	eBPF-powered network isolation for 5G use cases, balancing security and performance.
Farasat et al. [43] (XDP in Kubernetes)	DDoS Protection for Kubernetes Pods	eBPF, XDP, Weave Net VXLAN	Lightweight, robust DDoS mitigation using XDP programs.
Farasat et al. [44] (Intrusion Detection Datasets)	Intrusion Detection System (IDS) Dataset Generation	eBPF, XDP, Kubernetes testbed	Datasets for machine learning-driven intrusion detection in Kubernetes environments.

Koksal et al. [36] proposed a robust defense framework for mitigating DDoS attacks in Mobile Edge Computing (MEC) networks using Kubernetes. The approach incorporates scalable Containerized Network Functions (CNFs) with an Intrusion Detection and Prevention System (IDPS) to enhance adaptability and network security. This mechanism distributes resources across edge clusters, balancing the load on IDPS CNFs and effectively countering attacks like DNS floods and Yo-Yo. Kubernetes' auto-scaling capabilities allow for real-time adjustment of CNF deployments, meeting the lightweight, agile, and dynamic requirements of MEC environments. Validation through experiments in real MEC setups using OpenShift and Telco-grade profiles confirmed the system's efficiency in DDoS mitigation without significant resource overhead.

Miano et al. [6] proposed the eBPF as a flexible technology for advanced data processing within the Linux kernel. Initially used for monitoring tasks such as memory usage, Central Processing Unit (CPU) performance, page faults, and network traffic, eBPF has also shown potential for modifying data in transit. However, the development of complex network functions beyond simple proof-of-concept applications has proven challenging due to inherent limitations of the technology. Despite these challenges, eBPF remains promising, particularly due to unique features like dynamic recompilation of source code, which are not available in other solutions.

Miano et al. [37] analyzed approaches for integrating Smart Network Interface Cards (SmartNICs) into server-based data plane processing, focusing on DDoS mitigation. The proposed solution combines SmartNICs with technologies like eBPF/XDP to manage high traffic and DDoS attacks. A key feature of the solution is an adaptive hardware offloading mechanism that partitions traffic filtering between SmartNICs and the host, delegating the most aggressive DDoS sources to the SmartNIC. Experimental results showed that combining hardware filtering on the SmartNIC with XDP filtering on the host is the most efficient, offering better dropping rates and CPU usage.

The study found that while SmartNICs can reduce server load, they may not provide a complete solution for DDoS mitigation without additional components like a DDoS-aware load balancer. Additionally, reliance on SmartNIC CPUs alone for filtering can result in suboptimal performance due to their lower processing power.

Hohlfeld et al. [38] explored the advantages and limitations of the new offloading mechanisms available in the Linux kernel, focusing on generic AF XDP kernel-bypass, XDP device driver offloading, and offloading XDP programs to a Netronome SmartNIC. This study highlighted the challenges of offloading in virtualized environments, showing that while offloading can accelerate packet processing, it is most effective for small tasks. Overloading the SmartNIC with heavy tasks leads to performance degradation, and updating offloaded data can be expensive. The SmartNIC performed well in ultra-low latency processing for small workloads. Virtual machines benefit from offloading to the host, but care must be taken to ensure isolation and fairness, as offloading to the NIC may negatively affect the responsiveness of other VMs. This research concluded that while Linux's offloading framework holds significant potential, the actual benefits depend on the specific use case and require individual evaluation.

Liu et al. [39] proposed a non-intrusive network observability system for Kubernetes clusters using eBPF. The system collects L7/L4 protocol interaction data at the kernel level without requiring kernel or application code modifications, achieving over 10M throughput per second with less than 1% system impact. Machine learning methods are employed to diagnose network performance issues, identify bottlenecks, and localize problematic pods, enabling protocol-independent, efficient analysis of network performance in cloud-native environments.

Miano et al. [40] introduced Polycube, a software framework that brings Network Functions Virtualization (NFV) benefits to in-kernel packet processing. Unlike most NFV solutions that use kernel-bypass techniques, Polycube enables flexible, customizable network function chains within the kernel, combining efficient in-kernel data planes with user-space control planes. These "Cubes" can be dynamically created and injected into the kernel without custom modules, simplifying debugging and introspection. Polycube offers strong isolation, persistence, and composability features, crucial for cloud environments. The framework was validated through performance improvements and complex use cases, including a network provider for Kubernetes, demonstrating its versatility for cloud-native NFV applications.

Wang and Chang [41] designed an IDS that leverages eBPF in the Linux kernel for efficient packet inspection. Traditionally, IDS systems like Snort operate in user space, analyzing packet headers and payloads to detect intrusions. In their approach, the system is divided into two parts: the first, running in the kernel, uses eBPF to quickly pre-filter packets that are unlikely to match any detection rule. The second part, running in user space, examines the remaining packets for rule matches using a modified Snort ruleset. Experimental results showed that this eBPF-based IDS outperformed Snort by a factor of three in throughput under various conditions.

The study by Budigiri et al. [42] highlights the performance and security advantages of eBPF-powered solutions like Calico and Cilium, which enforce dynamic, low-overhead network isolation between containers. These tools meet the stringent demands of 5G edge-computing use cases by balancing robust security with minimal latency, challenging the misconception that securing inter-container communication compromises performance.

Farasat et al. [43] demonstrated the effectiveness of eBPF/XDP in safeguarding Kubernetes Pods by employing XDP programs (e.g., XDP_DROP/FILTER) over the Weave Net VXLAN interface. This approach offered a lightweight yet robust mechanism to prevent DDoS attacks from rendering Pods or nodes inaccessible. In another study, Farasat et al. [44] examined the integration of eBPF/XDP in a Kubernetes-based testbed to generate intrusion detection datasets, modeling both malicious and normal traffic. This dataset supports machine learning frameworks, enabling advancements in intrusion detection systems for Kubernetes environments.

The reviewed studies can be categorized into two primary groups based on their objectives and methodologies:

a) eBPF in intrusion detection and complex network monitoring. This category includes studies such as [6], [41], [45], which emphasize leveraging eBPF for kernel-level event monitoring and real-time packet filtering

Table 2: Enhanced comparative analysis of DDoS mitigation techniques.

Technique	Mechanism	Advantages	Limitations	Applicability in Kubernetes
Rate Limiting	Throttling traffic based on thresholds	Simple, lightweight, and effective for volumetric attacks	May block legitimate traffic, struggles with adaptive attacks	Suitable for basic API rate control and ingress throttling
Traffic Filtering	Rule-based packet inspection	High precision, customizable filtering	High computational overhead for complex rules	Can be optimized with eBPF for scalable filtering
Behavioral Anomaly Detection	AI/ML-based traffic analysis	Identifies zero-day and adaptive threats	Prone to false positives, requires continuous training	Enhances IDS/IPS for dynamic Kubernetes workloads
Signature-Based Detection	Pattern matching against known attack signatures	Effective against known threats	Fails against zero-day and evolving attack techniques	Complements other techniques in multi-layer security
Flow-Based Monitoring	Analyzing traffic flow metrics	Detects volumetric and slow-rate attacks	Performance impact in high-throughput environments	Helps in proactive attack detection and forensic analysis
eBPF/XDP Integration	Kernel-level packet filtering	Ultra-low latency, efficient at scale, programmable	Limited by kernel version dependencies	Optimized for Kubernetes-native security enforcement

b) DDoS mitigation and kernel bypassing for performance optimization. This category includes studies such as [35], [37], which focus on the implementation of eBPF to enhance intrusion detection systems (IDS) by capturing and analyzing data directly within the kernel.

2.2. Comparative Analysis of Various DDoS Mitigation Techniques in Kubernetes Environments

Table 2 provides a comparative analysis of various DDoS mitigation techniques, highlighting their mechanisms, strengths, limitations, and specific applicability in Kubernetes environments.

Rate limiting, for instance, offers a simple and effective way to control incoming traffic volume, yet it risks blocking legitimate traffic, especially in dynamic containerized applications where patterns can fluctuate unpredictably.

Traffic filtering, on the other hand, relies on predefined rules to inspect packets, ensuring high precision but introducing computational overhead in large-scale Kubernetes deployments - a limitation that eBPF mitigates through efficient rule enforcement.

Behavior anomaly detection emerges as a robust strategy for identifying unusual traffic patterns, leveraging advanced statistical or machine learning models. This approach excels in Kubernetes clusters with variable workload behavior but often suffers from false positives, which could disrupt legitimate service flows.

Lastly, the integration of eBPF and XDP provides a kernel-level solution with unparalleled performance and low latency, combining the scalability of eBPF with the efficiency of XDP.

In real-world deployments, the choice between behavioral anomaly detection and signature-based detection depends on several factors, such as the nature of the network traffic, system performance requirements, and the specific security threat landscape. While signature-based detection is highly effective for identifying known attacks, it may struggle to detect novel or zero-day threats. On the other hand, behavioral anomaly detection, although capable of identifying previously unseen attack patterns, requires a baseline of normal behavior and may result in a higher rate of false positives, particularly in dynamic environments like Kubernetes. Furthermore, eBPF and XDP provide powerful solutions for packet filtering and DDoS mitigation but come with trade-offs, particularly when deployed in resource-constrained environments or when the kernel becomes overwhelmed. For instance, as network traffic increases or the number of monitored pods grows, XDP may experience inefficiencies due to kernel limitations, such as memory bottlenecks or reduced packet processing speed. In such cases, alternative methods or

Table 3: Features of eBPF and XDP in network security.

Feature	eBPF	XDP	Combined use in Kubernetes
Packet filtering	User-space	Kernel-space	High-speed filtering for DDoS prevention
Resource efficiency	Moderate	High	Low overhead for containerized environments
Scalability	High	High	Optimal for dynamic Kubernetes clusters
Observability	Extensive	Limited	Enhanced monitoring with eBPF tracing

optimizations, such as offloading some processing tasks to SmartNICs or using hybrid models, may offer better scalability and performance without compromising security. By considering these factors, organizations can make informed decisions about which detection methods and technologies to implement based on their specific needs and deployment scenarios.

Table 3 outlines the key features of eBPF and XDP, comparing their individual capabilities and their combined potential in enhancing network security within Kubernetes environments. Packet filtering, a critical function for DDoS mitigation, is handled differently by the two technologies: eBPF operates in user space, offering flexibility for detailed traffic analysis, while XDP processes packets in kernel space, providing unmatched speed and efficiency. Together, they enable high-performance, low-latency filtering suitable for Kubernetes clusters with high traffic demands. Resource efficiency also varies between the two approaches. While eBPF exhibits moderate resource consumption, XDP is designed for high efficiency, making it ideal for large-scale, latency-sensitive applications. When used together, they ensure a lightweight solution tailored to the dynamic nature of containerized workloads. Additionally, scalability is a shared strength, with both technologies capable of handling the demands of dynamic and distributed Kubernetes deployments. In terms of observability, eBPF shines with its extensive tracing and monitoring capabilities, allowing for deep insights into system and network behavior. XDP, being focused on high-speed packet processing, has more limited observability features. However, when combined, eBPF's tracing complements XDP's performance, enabling robust monitoring and analysis tools essential for maintaining security and performance in Kubernetes environments.

Table 4 identifies the key challenges in implementing eBPF for DDoS mitigation and shows strategies to address these issues, ensuring effective deployment within Kubernetes environments.

The complexity of kernel programming is a primary concern, as eBPF requires significant expertise to develop and optimize. To mitigate this, frameworks like BCC and libbpf provide abstractions that simplify development, enabling broader adoption among security engineers.

Performance tuning poses another challenge, as eBPF programs must be carefully optimized to minimize latency, particularly in real-time applications. Regular benchmarking and iterative optimization during development are crucial to address this, ensuring that eBPF solutions meet the high-performance demands of Kubernetes-based deployments.

Compatibility with Kubernetes represents a critical consideration, particularly given the inherent complexity associated with achieving seamless integration within containerized and microservices-based environments. The orchestration of security policies in such dynamic ecosystems necessitates advanced tooling capable of interfacing natively with Kubernetes constructs. In this context, plugins such as Cilium serve a pivotal role by offering robust, Kubernetes-native support for eBPF, thereby facilitating the efficient deployment, management, and enforcement of security policies at scale. Through this integration, Cilium not only abstracts much of the underlying complexity but also enhances observability, performance, and policy granularity within cloud-native security architectures.

Security risks, including potential exploitation of eBPF programs, must also be addressed to safeguard the system. Implementing robust verification and sandboxing mechanisms is critical to ensure that only validated and secure programs are deployed in production environments.

Table 4: Challenges and mitigation strategies in eBPF implementation.

Challenge	Description	Mitigation Strategy
Complexity of Kernel programming	High expertise required for eBPF/XDP development	Use frameworks like BCC or libbpf for simplified programming
Performance tuning	Optimizing eBPF programs for latency-sensitive applications	Benchmark and optimize during development
Compatibility with Kubernetes	Ensuring seamless integration with container orchestration	Use plugins like Cilium for Kubernetes-native support
Security risks	eBPF programs could be exploited if not secured properly	Implement strict verification and sandboxing mechanisms

Table 5 outlines key metrics for evaluating the performance and effectiveness of eBPF-based DDoS mitigation solutions, providing a comprehensive framework to assess their applicability in Kubernetes environments. These metrics cover various aspects of eBPF's performance, such as efficiency, resource utilization, and detection accuracy, and help in determining how well the technology mitigates DDoS attacks without compromising the performance or reliability of containerized environments.

Throughput is a critical metric when assessing eBPF-based DDoS mitigation solutions. It refers to the number of packets that can be processed per second by the system, offering an insight into how effectively the solution can handle large-scale traffic, especially during a DDoS attack. For Kubernetes environments, where traffic patterns can be unpredictable and fluctuate rapidly, high throughput ensures that the mitigation system can manage the volume of incoming packets without introducing bottlenecks. A solution with higher throughput is especially important when mitigating volumetric DDoS attacks that aim to overwhelm network resources.

Latency is another essential metric that measures the delay introduced by packet processing. In real-time mitigation scenarios, especially within high-speed containerized environments like Kubernetes, low latency is paramount to ensure that malicious packets are dropped or redirected before they can reach critical resources. High latency could result in delayed detection and mitigation, leaving the system vulnerable to DDoS attacks. In Kubernetes, where services are often distributed and dynamic, high latency could also disrupt service availability, making it a critical metric to monitor.

Resource utilization measures the CPU and memory usage of eBPF programs during packet processing. Since Kubernetes environments often rely on microservices that are lightweight and dynamically scalable, maintaining low overhead is vital. eBPF, known for its efficiency, operates within the kernel and avoids the need for context switching, which reduces its overall resource consumption. This ensures that eBPF-based mitigation solutions can run efficiently on containerized platforms, preventing resource exhaustion and minimizing the impact on other containerized applications. A low resource utilization metric indicates that eBPF-based solutions are optimized for real-time, high-throughput environments like Kubernetes.

Detection accuracy refers to the proportion of malicious packets that are correctly identified by the mitigation solution. High detection accuracy ensures that a significant number of malicious traffic patterns are identified, reducing the chances of a DDoS attack succeeding. However, it’s important to balance detection accuracy with the false positive rate, which will be discussed next. In the context of Kubernetes, detection accuracy becomes particularly important for identifying sophisticated, low-and-slow DDoS attacks that may be missed by less advanced filtering methods.

False positive rate is a critical metric in evaluating the precision of eBPF-based DDoS mitigation solutions. It measures the rate at which legitimate traffic is incorrectly flagged as malicious. A high false positive rate can result in blocking or throttling legitimate users, leading to service disruptions and a poor user experience. Given the dynamic nature of Kubernetes workloads, where application traffic patterns can vary greatly, minimizing false positives is essential. Advanced anomaly detection algorithms integrated into eBPF-based systems can help reduce false positives by distinguishing between benign traffic and malicious patterns more effectively.

Table 5: Metrics for evaluating eBPF-based DDoS mitigation solutions

Metric	**Definition**	**Relevanceto DDoS mitigation**
Throughput	Number of packets processed per second	Measures efficiency during high traffic loads
Latency	Delay introduced by packet processing	Critical for real-time mitigation
Resource utilization	CPU and memory usage of eBPF programs	Ensures lightweight implementation
Detection accuracy	Proportion of malicious packets correctly identified	Determines the reliability of the solution
False positive rate	Rate of legitimate traffic incorrectly flagged	Indicates precision in anomaly detection

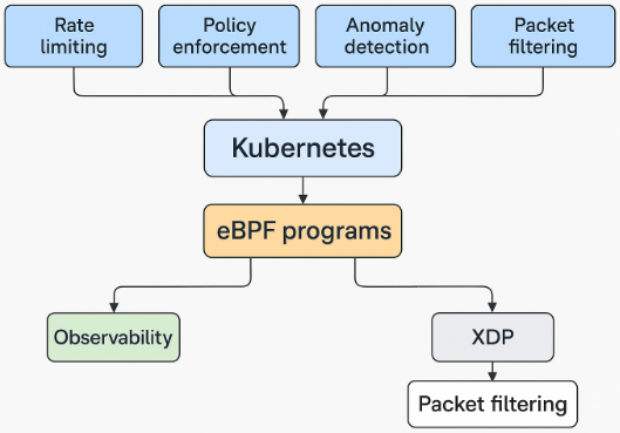

Fig. 1: The integration of Kubernetes, eBPF, and XDP for real-time DDoS detection and mitigation.

Empirical studies provide concrete evidence of the performance benefits associated with eBPF and XDP in mitigating Distributed Denial-of-Service (DDoS) attacks within Kubernetes environments. XDP, in particular, achieves ultra-low latency - often in the range of tens of microseconds - by intercepting and processing packets at the earliest possible stage in the Linux kernel's networking stack. This bypasses significant portions of the conventional networking path, making XDP especially well-suited for high-throughput, real-time DDoS mitigation scenarios [42]. In contrast, eBPF offers a higher degree of programmability and supports integration with user-space applications, which introduces slightly greater resource demands. Nevertheless, eBPF consistently maintains low latency, typically below 1 millisecond, while enabling deep packet inspection and complex traffic anomaly detection mechanisms [40]. These performance characteristics underline the practical viability of both technologies for securing containerized environments against increasingly sophisticated network-layer attacks.

Fig. 1 shows the integration of Kubernetes, eBPF, and XDP for real-time DDoS detection and mitigation. It aligns with established practices in the field, where XDP operates at the earliest point in the kernel's networking stack to provide ultra-low latency packet processing, and eBPF offers programmable flexibility for traffic inspection and anomaly detection within Kubernetes environments. This integration facilitates efficient and scalable DDoS mitigation strategies, as corroborated by empirical studies and industry implementations.

2.3. Future Directions and Research Opportunities

Novel approaches and future opportunities for enhancing eBPF-based DDoS mitigation strategies in Kubernetes environments are outlined below. These explore gaps in current research and introduce innovative applications aimed at securing and scaling containerized platforms. Each category introduces potential innovations that could significantly improve the effectiveness of eBPF as a security mechanism within Kubernetes environments, such as:

a) Packet Filtering. The current approach to DDoS mitigation in Kubernetes environments primarily relies on kernel-level eBPF filtering. Techniques such as the XDP_DROP program provide a robust foundation for high-performance packet filtering directly within the kernel. This method enables rapid packet handling and early-stage threat mitigation. However, emerging innovations propose the integration of AI-driven adaptive filtering with eBPF programs. This integration represents a significant advancement. By leveraging real-time traffic analysis, the filtering strategy can dynamically adjust in response to evolving attack patterns. Such adaptability enhances the precision

of mitigation efforts, reduces the rate of false positives, and improves the responsiveness of security mechanisms. As DDoS attacks become increasingly sophisticated and diverse, this context-aware, intelligent filtering becomes essential for maintaining effective and resilient network defenses.

b) Observability. Network observability in Kubernetes environments is critical for proactively identifying and mitigating potential security threats. Current methods primarily rely on static eBPF programs for monitoring network traffic. The proposed innovation of integrating dynamic, event-driven eBPF observability frameworks powered by machine learning could drastically improve this aspect. Such an approach would enable real-time detection of anomalous patterns, providing not only reactive but also proactive capabilities. The ability to detect evolving threats and abnormal behavior in real time would enhance the overall security posture, enabling Kubernetes administrators to act swiftly before attacks escalate.

c) Microservices security. In Kubernetes environments, where microservices architecture is prevalent, securing internal communication between containers is a critical concern. The existing focus is on application-level security policies, often implemented via eBPF/XDP. The proposed innovation introduces eBPF-based inter-container anomaly detection, utilizing behavioral profiling of microservices. This innovation could help detect deviations from normal microservice behavior, preventing lateral movement of threats within the cluster. By focusing on microservices-specific security, this innovation would bolster internal defense mechanisms, making it harder for adversaries to escalate their attacks or propagate them within the environment.

d) DDoS dataset creation. The generation of realistic datasets for training intrusion detection models is essential for enhancing the accuracy and reliability of security systems. While current research focuses on synthetic and controlled traffic generation through testbeds, the proposed approach of integrating live Kubernetes cluster telemetry with attack emulation holds promise for creating more realistic datasets. This innovation would bridge the gap between theoretical testbeds and real-world scenarios, providing intrusion detection models with practical, real-world data. As a result, security solutions could be better equipped to handle the complexities of DDoS attacks in production Kubernetes environments.

e) Edge computing. The shift toward edge computing and 5G environments introduces new security challenges, particularly concerning the latency-sensitive nature of applications in these settings. While current DDoS mitigation strategies are mainly tailored for centralized cloud environments, the proposed innovation aims to optimize eBPF-based DDoS protection specifically for edge computing. By adapting eBPF to the unique needs of edge and 5G environments, this approach would improve the reliability and performance of applications at the network edge, ensuring robust protection against DDoS attacks in decentralized, low-latency contexts. This is crucial as edge computing becomes increasingly central to modern cloud-native infrastructures.

f) Policy enforcement. Policy enforcement in Kubernetes environments is typically handled through static tools such as Open Policy Agent (OPA) and Gatekeeper. The proposed shift to AI-driven, adaptive policy enforcement using eBPF could revolutionize security policy management in Kubernetes. This approach would allow for fine-grained, context-aware security policies that dynamically adapt to emerging threats, reducing the need for manual intervention and improving overall security compliance. The use of AI would enable more flexible and responsive policy enforcement, ensuring that Kubernetes environments remain secure as they scale and evolve.

g) Energy efficiency. The growing emphasis on sustainability and energy efficiency in cloud infrastructures makes the energy consumption of security mechanisms an important consideration. Current eBPF programs primarily focus on performance-centric goals, particularly in DDoS mitigation. The proposed innovation introduces energy-efficient eBPF algorithms that prioritize power-saving modes within Kubernetes environments. This would not only reduce energy consumption but also align with the broader goals of creating sustainable, green cloud infrastructures. As the demand for energy-efficient solutions rises, this innovation could provide a significant step toward minimizing the carbon footprint of Kubernetes security solutions.

h) Multi-cloud security. Security across hybrid and multi-cloud environments is a key challenge as organizations increasingly adopt distributed cloud architectures. While current eBPF-based security solutions typically

focus on single-cluster Kubernetes security, the proposed innovation aims to extend eBPF capabilities to cross-cloud Kubernetes environments. By leveraging distributed eBPF telemetry and threat intelligence, this approach would enable seamless, consistent security across hybrid and multi-cloud setups. This innovation is particularly important as organizations seek to maintain robust, unified security policies across diverse cloud infrastructures, ensuring effective DDoS mitigation in complex environments.

The proposed future directions and research opportunities outlined in the table are firmly justified by practical technologies that enhance network security and are aligned with recommendations from the scientific literature. On the practical side, these innovations address real-world challenges observed in deploying eBPF-based DDoS mitigation strategies within Kubernetes environments, where scalability, adaptability, and precision in security measures remain critical. These challenges have been highlighted in industry reports and real-world case studies, thus underscoring the need for dynamic, AI-integrated solutions and cross-layer security mechanisms. From a scientific literature standpoint, the proposed innovations are deeply informed by recent advancements and calls for further exploration within the field, as reflected in the analyzed studies.

3. Conclusion

This review has explored the transformative role of eBPF and XDP technologies in enhancing network security, particularly in mitigating DDoS attacks within Kubernetes environments. By leveraging these advanced tools, we demonstrated their potential for high-performance, kernel-level packet filtering and intrusion detection, ensuring scalable and efficient protection for containerized applications. The integration of eBPF with Kubernetes allows for real-time observability and microservice-level security without compromising system performance, addressing critical challenges posed by modern cloud-native infrastructures. This research also highlighted the synergy between eBPF and existing Kubernetes security solutions, including network policies and runtime enforcement tools, to deliver a comprehensive defense against evolving threats. Through an analysis of contemporary approaches, we identified the advantages of eBPF in providing low-latency and resource-efficient DDoS mitigation techniques, particularly in complex microservices architectures. Our findings emphasize the need for further innovation in combining eBPF with machine learning models and automated security frameworks to preemptively address emerging vulnerabilities. By aligning with the scalability and dynamic resource management capabilities of Kubernetes, the proposed solutions ensure the resilience and stability of distributed systems. Future work should extend these advancements to encompass broader use cases, such as securing multi-cloud environments and improving data plane observability. Additionally, exploring the integration of eBPF with other cloud-native security tools, such as service meshes, could further enhance protection against advanced persistent threats (APTs) and other sophisticated attack vectors. Finally, research into the optimization of eBPF's resource consumption will be crucial for its widespread adoption in large-scale, resource-constrained environments.

CRediT Authorship Contribution Statement

Mircea Ţălu: Writing – Review & Editing, Writing – Original Draft, Validation, Resources, Software, Methodology, Investigation, Formal analysis, Data Curation, Conceptualization, Project Administration.

Declaration of Competing Interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

All relevant data are within the manuscript and its supporting information files.

Declaration of Generative AI and AI-assisted Technologies in The Writing Process

The authors used generative AI to improve the writing clarity of this paper. They reviewed and edited the AI-assisted content and take full responsibility for the final publication.

References

[1] C. Kaufman, R. Perlman, M. Speciner, R. Perlner. *Network security: private communication in a public world*, 3rd ed. Pearson Education, Hoboken, NJ, USA, 2022.

[2] A. Sadiq, H.J. Syed, A.A. Ansari, A.O. Ibrahim, M. Alohaly, M. Elsadig, "Detection of Denial of Service attack in cloud-based Kubernetes using eBPF," *Applied Sciences*, vol. 13, no. 8, p. 4700, 2023, doi: 10.3390/app13084700.

[3] S. McCanne, V. Jacobson, "The BSD Packet Filter: A New Architecture for User-Level Packet Capture," in *Proceedings of the USENIX Winter*, San Diego, CA, USA, 25–29 January 1993, vol. 46, doi: 10.5555/1267303.1267305.

[4] B. Gregg, *BPF performance tools: Linux system and application observability*, 1st ed., Addison-Wesley Professional, Boston, USA, 2019.

[5] G. Ognibene, "Master's thesis: Toward efficient DDoS detection with eBPF," Politecnico di Torino, Turin, Italy, 2021.

[6] S. Miano, M. Bertrone, F. Risso, M. Tumolo, M.V. Bernal, "Creating complex network services with eBPF: experience and lessons learned," *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*, Bucharest, Romania, 2018, pp. 1-8, doi: 10.1109/HPSR.2018.8850758.

[7] D. Scholz, D. Raumer, P. Emmerich, A. Kurtz, K. Lesiak, G. Carle, "Performance implications of packet filtering with Linux eBPF," in *Proceedings of the 2018 30th International Teletraffic Congress (ITC 30)*, Vienna, Austria, 3–7 September 2018, vol. 1, pp. 209–217, doi: 10.1109/itc30.2018.00039.

[8] H.J. Hadi, M. Adnan, Y. Cao, F.B. Hussain, N. Ahmad, M.A. Alshara, Y. Javed, "iKern: Advanced intrusion detection and prevention at the Kernel level using eBPF," *Technologies*, vol. 12, no. 8, p. 122, 2024, doi: 10.3390/technologies12080122.

[9] J. Gallego-Madrid, R. Sanchez-Iborra, A.S. Gomez, "eBPF and XDP technologies as enablers for ultra-fast and programmable next-gen network infrastructures," in *A. Mukherjee, D. De, R. Buyya (eds.), Resource Management in Distributed Systems, Studies in Big Data*, vol. 151, Springer, Singapore, 2024, doi: 10.1007/978-981-97-2644-8_13.

[10] A. Hohn, *The book of Kubernetes: a complete guide to container orchestration*, 1st ed., No Starch Press, San Francisco, CA, USA, 2022.

[11] L. Rice, M. Hausenblas, *Kubernetes security: operating Kubernetes clusters and applications safely*, O'Reilly Media, Sebastopol, CA, USA, 2018.

[12] S. Lee, J. Nam, "Kunerva: automated network policy discovery framework for containers," in *IEEE Access*, vol. 11, pp. 95616-95631, 2023, doi: 10.1109/ACCESS.2023.3310281.

[13] "Security tools for containers, Kubernetes, and cloud," Aug. 2023. Available: https://sysdig.com.

[14] "Cloud-Native Security Tool Designed for Linux Systems," Aug. 2023. Available: https://falco.org/.

[15] C.-W. Tien, T.-Y. Huang, C.-W. Tien, T.-C. Huang, S.-Y. Kuo, "KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches," *Engineering Reports*, vol. 1, no. 5, p. e12080, 2019, doi: 10.1002/eng2.12080.

[16] "Cloud-Native Runtime Security Enforcement System," Aug. 2023. Available: https://kubearmor.io/.

[17] "Policy-Based Control for Cloud Native Environments," Aug. 2023. Available: https://www.openpolicyagent.org/.

[18] "Policy Controller for Kubernetes," Aug. 2023. Available: https://github.com/open-policy-agent/gatekeeper.

[19] A. Bremler-Barr, M. Czeizler, H. Levy, J. Tavori, "Exploiting miscoordination of microservices in tandem for effective DDoS attacks," *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, Vancouver, BC, Canada, 2024, pp. 231-240, doi: 10.1109/INFOCOM52122.2024.10621335.

[20] J.J. Lawrence, E. Prakash, C. Hewage, "Securing Kubernetes: a study on the measures for enhancing control and data plane security," in *C. Hewage, L. Nawaf, N. Kesswani (eds.), AI applications in cyber security and communication networks, ICCS 2023, Lecture Notes in Networks and Systems*, vol. 1032, Springer, Singapore, 2024, doi: 10.1007/978-981-97-3973-8_9.

[21] R.M.A. Haseeb-ur-rehman, A.H.M. Aman, M.K. Hasan, K.A.Z. Ariffin, A. Namoun, A. Tufail, K-H. Kim, "High-speed network DDoS attack detection: a survey," *Sensors*, vol. 23, no. 15, p. 6850, 2023, doi: 10.3390/s23156850.

[22] M. Driss, D. Hasan, W. Boulila, J. Ahmad, "Microservices in IoT security: current solutions, research challenges, and future directions," *Procedia Computer Science*, vol. 192, pp. 2385-2395, 2021, doi: 10.1016/j.procs.2021.09.007.

[23] W. Lee, Y.R. Choe, R.S. Ghosh, "Recurrent neural network and convolutional neural network for detection of Denial of Service attack in microservices," *2023 International Conference on Machine Learning and Applications (ICMLA)*, Jacksonville, FL, USA, 2023, pp. 1451-1456, doi: 10.1109/ICMLA58977.2023.00219.

[24] R.K. Jayalath, H. Ahmad, D. Goel, M.S. Syed, F. Ullah, "Microservice vulnerability analysis: a literature review with empirical insights," in *IEEE Access*, vol. 12, pp. 155168-155204, 2024, doi: 10.1109/ACCESS.2024.3481374.

[25] M. Țălu, "A review of vulnerability discovery in WebAssembly binaries: insights from static, dynamic, and hybrid analysis," *Acta Technica Corviniensis – Bulletin of Engineering*, Hunedoara, Romania, Tome XVII, Fascicule 4, pp. 13-22, 2024.

[26] M. Țălu, "A review of advanced techniques for data protection in WebAssembly," *Annals of Faculty of Engineering Hunedoara, International Journal of Engineering*, Hunedoara, Tome XXII, Fascicule 4, pp. 131-136, 2024.

[27] M. Țălu, "Security and privacy in the IIoT: threats, possible security countermeasures, and future challenges," *Computing&AI Connect*, vol. 2, Article ID: 2025.0011, 2025, doi: 10.69709/CAIC.2025.139199.

[28] M. Țălu, "A comparative study of WebAssembly runtimes: performance metrics, integration challenges, application domains, and security features," *Archives of Advanced Engineering Science*, 2025, doi: 10.47852/bonviewAAES52024965.

[29] Ş. Țălu, "Strategic measures in improving cybersecurity management in micro and small enterprises," in *2nd International Scientific and Practical Conference on Digital Economy (ISCDE 2020)*, 2020, pp. 522–528, doi: 10.2991/aebmr.k.201205.087.

[30] L.-D. Chou, L.-Y. Jian, Y.-W. Chen, "eBPF-based network monitoring platform on Kubernetes," in *2024 6th International Conference on Computer Communication and the Internet (ICCCI)*, Tokyo, Japan, 2024, pp. 140-144, doi: 10.1109/ICCCI62159.2024.10674074.

[31] W. Yang, P. Chen, G. Yu, H. Zhang, H. Zhang, "Network shortcut in data plane of service mesh with eBPF," *Journal of Network and Computer Applications*, vol. 222, Article 103805, 2024, doi: 10.1016/j.jnca.2023.103805.

[32] J. Nam, S. Lee, P. Porras, V. Yegneswaran, S. Shin, "Secure inter-container communications using XDP/eBPF," *IEEE/ACM Transactions on Networking*, vol. 31, no. 2, pp. 934-947, 2023, doi: 10.1109/TNET.2022.3206781.

[33] I. Riadi, R. Umar, A. Sugandi, "Web forensic on Kubernetes cluster services using GRR rapid response framework," *Int. J. Sci. Technol. Res.*, vol. 9, pp. 3484–3488, 2020, doi: 10.15294/sji.v7i1.18299.

[34] A. Modak, S. D. Chaudhary, P. S. Paygude, S. R. Ldate, “Techniques to Secure Data on Cloud: Docker Swarm or Kubernetes?,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 2018, pp. 7-12, doi: 10.1109/ICICCT.2018.8473104.

[35] G. Bertin, “XDP in practice: integrating XDP into our DDoS mitigation pipeline,” in *Proceedings of the Technical Conference on Linux Networking, Netdev*, Montréal, QC, Canada, 6–8 April 2017, vol. 2.

[36] S. Koksal, F.O. Catak, Y. Dalveren, “Flexible and lightweight mitigation framework for Distributed Denial-of-Service attacks in container-based edge networks using Kubernetes,” *IEEE Access*, vol. 12, pp. 172980-172991, 2024, doi: 10.1109/ACCESS.2024.3501192.

[37] S. Miano, R. Doriguzzi-Corin, F. Risso, D. Siracusa, R. Sommese, “Introducing smartnics in server-based data plane processing: The DDoS mitigation use case,” *IEEE Access*, vol. 7, pp. 107161–107170, 2019, doi: 10.1109/ACCESS.2019.2933491.

[38] O. Hohlfeld, J. Krude, J.H. Reelfs, J. Ruth, K. Wehrle, “Demystifying the Performance of XDP BPF,” in *Proceedings of the 2019 IEEE Conference on Network Softwarization (NetSoft)*, Paris, France, 24–28 June 2019, doi: 10.1109/NETSOFT.2019.8806651.

[39] C. Liu, Z. Cai, B. Wang, Z. Tang, J. Liu, “A protocol-independent container network observability analysis system based on eBPF,” in *Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, Hong Kong, China, 2–4 December 2020, pp. 697–702.

[40] S. Miano, F. Risso, M.V. Bernal, M. Bertrone, Y. Lu, “A framework for eBPF-based network functions in an era of microservices,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, pp. 133–151, 2021, doi: 10.1109/TNSM.2021.3055676.

[41] S.-Y. Wang, J.-C. Chang, “Design and implementation of an intrusion detection system by using Extended BPF in the Linux kernel,” *Journal of Network and Computer Applications*, vol. 198, Article 103283, 2022, doi: 10.1016/j.jnca.2021.103283.

[42] G. Budigiri, C. Baumann, J.T. Mühlberg, E. Truyen, W. Joosen, “Network policies in Kubernetes: performance evaluation and security analysis,” in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Porto, Portugal, 2021, pp. 407-412, doi: 10.1109/EuCNC/6GSummit51104.2021.9482526.

[43] T. Farasat, M.A. Rathore, J. Kim, “Securing Kubernetes Pods communicating over Weave Net through eBPF/XDP from DDoS attacks,” in *CODASPY '23: Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, pp. 287-289, 2023, doi: 10.1145/3577923.3585049.

[44] T. Farasat, J.W. Kim, J. Posegga, “Advancing network security: a comprehensive testbed and dataset for machine learning-based intrusion detection,” *arXiv:2410.18332v1*, 2024, doi: 10.48550/arXiv.2410.18332.

[45] M. Abranches, O. Michel, E. Keller, S. Schmid, “Efficient network monitoring applications in the kernel with eBPF and XDP,” in *Proceedings of the 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Heraklion, Greece, 9–11 November 2021, pp. 28–34, doi: 10.1109/NFVSDN53031.2021.9665095.