electronics

MDPI

Article

InDepth: A Distributed Data Collection System for Modern Computer Networks

Angel Kodituwakku * and Jens Gregor

Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, TN 37996, USA; jgregor@utk.edu
* Correspondence: hkoditu1@alum.utk.edu

Abstract: Cybersecurity researchers and security analysts rely heavily on data to train and test network threat detection models, and to conduct post-breach forensic analyses. Comprehensive data-including network traces, host telemetry, and contextual information-are crucial for these tasks. However, widely used public datasets often suffer from outdated network traffic and features, statistical anomalies, and simulation artifacts. Furthermore, existing data collection systems frequently face architectural and computational limitations, necessitating workarounds that result in incomplete or disconnected data. Currently, no framework provides comprehensive data collection from all network segments without requiring specialized or proprietary hardware or software agents. This paper introduces InDepth, a scalable system employing a distributed, data-link layer architecture that enables comprehensive data acquisition across entire networks. We also present a model cyber range capable of dynamically generating datasets for evaluation. We demonstrate the effectiveness of InDepth using real-world network data.

Keywords: cybersecurity; data collection; cyber range; incident response; network telemetry

check for updates

Academic Editor: Domenico Ursino

Received: 31 March 2025
Revised: 1 May 2025
Accepted: 7 May 2025
Published: 12 May 2025

Citation: Kodituwakku, A.; Gregor, J. InDepth: A Distributed Data Collection System for Modern Computer Networks. *Electronics* **2025**, *14*, 1974. https://doi.org/10.3390/electronics14101974

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Cybersecurity researchers developing machine learning models for threat detection and security analysts conducting forensic investigations both rely heavily on high-quality data. The effectiveness of predictive models and the precision of post-breach analyzes depend on data sets that accurately reflect real-world network traffic scenarios and provide sufficient context to understand security incidents.

Widely used benchmark datasets, such as DARPA'98 [1], KDD'99 [2], and NSL-KDD [3], are outdated and do not represent modern network traffic [4]. These datasets often contain simulation artifacts and statistical anomalies, which can cause machine learning models to achieve artificially high accuracy scores that cannot be replicated in real world data [5]. Furthermore, they typically lack crucial contextual information about the data collection environment, including network topology, subnet functions, firewall rules, host operating systems, service versions, and specific attack details. Validation of the accuracy of the represented traffic types is also often missing. Another significant limitation is the practice of labeling individual flow records as simply benign or malicious [6]. This approach encourages models to classify threats based on single flows, which is insufficient because individual flows rarely constitute a complete attack. Effective forensic analysis, in contrast, requires examining the broader context, including victim vulnerabilities, attacker motivations, and the sequence of actions leading to a compromise.

Consequently, there is a clear need for accurate and comprehensive datasets that capture the dynamics of contemporary threats within modern networks [7]. Creating

such datasets is challenging because current data collection systems (DCSs) often rely on workarounds for architectural or computational limitations. These workarounds can result in incomplete or disconnected host and network activity data, potentially missing crucial information needed to detect and analyze sophisticated attacks such as reconnaissance and lateral movement [8]. Relying solely on flow records is also inadequate, as the data within them can be misleading; for instance, source addresses might be spoofed, or traffic could be relayed through unsuspecting hosts. Therefore, a robust DCS is required that is capable of capturing all network interactions, including intra- and inter-VLAN traffic, and is open to deployment in diverse real-world networks, physical or virtual [7]. In addition to network traces, such a system must collect host telemetry and contextual information, ensuring that the data are reproducible across a variety of network configurations and for different traffic conditions.

This paper presents InDepth, a scalable DCS that leverages a distributed architecture at the data-link layer to facilitate the needed data collection. In addition, a model cyber range capable of dynamically generating datasets is introduced. The motivation for the work stems from analysis of data from the Western Regional Collegiate Cyber Defense Competition (WRCCDC) 2020 competition [9]. Said data are also used to evaluate effectiveness of the approach. A companion paper that describes an endpoint detection system called InMesh provides a showcase application [10].

The literature divides intrusion detection systems into two main categories, namely, host-based (HIDS) and network-based (NIDS). Installed on individual hosts, HIDS monitors log files and user activity. In contrast, NIDS passively monitors network traffic to identify known threat signatures or anomalous behavior. HIDS is intrusive as a result of running on the devices but provides access to a wide range of information about the hosts. The approach may not be feasible in certain situations depending on the security policies of the network, the host operating systems, and the privacy concerns of the user. Although InDepth is NIDS-like, we show that host attributes can be collected without the use of software agents. These attributes can be used to observe host behavior over time.

The remainder of the paper is organized as follows. Section 2 summarizes benchmark datasets and network data collection systems. Section 3 develops the InDepth system architecture along with the cyber range mentioned. Section 4 provides experimental results based on the WRCCDC 2020 data for three different scenarios. Section 5 provides closing remarks.

2. Background

2.1. Benchmark Datasets

Several benchmark datasets have been developed, notably three from the MIT Lincoln Lab, DARPA'98, which contains seven weeks of synthetic tcpdump data with injected attacks, and its derivatives, KDD'99 and NSL-KDD [11]. KDD'99 remains widely used, in part due to its availability through platforms such as TensorFlow to develop and test machine learning-based intrusion detection algorithms [12].

KDD'99 has many shortcomings. McHugh pointed out that the data do not represent commodity network traffic and noted a lack of documentation even for benign activity [4]. Kayacik et al. reported extreme class imbalance, with 98% of training data belonging to only three classes (Normal, Neptune DoS, Smurf DoS), leading to inflated performance metrics for these common types [13]. Tavallaee et al. discovered substantial record duplication (78% in training, 76% in testing), which biases detection algorithms towards frequent records and potentially causes them to miss less common, but potentially harmful attacks [3]. They also found that easily detectable DoS attacks constitute more than 71% of the attack data. The high accuracy often reported for KDD'99 is largely attributed to simulation artifacts

and statistical anomalies rather than realistic detection capabilities [3,5]. With respect to how the data were generated, use of the single-threaded 'tcpdump' tool raises concerns about potential packet drops, an aspect not thoroughly investigated. Additionally, some of the features included were derived from cleartext packet content (e.g., login attempts, privilege escalation), which cannot be performed in modern networks due to widespread encryption (e.g., SSH). Although host agents could potentially collect such data, many associated attacks (such as Smurf, which is now obsolete) are often preventable by standard firewalls and system configurations without requiring complex detection algorithms. On a final note, the labeling scheme is problematic. Records are simply tagged as 'normal' or one of four attack types, lacking crucial context like victim OS/service versions or attacker tools and parameters. Moreover, labeling individual flows as attacks is insufficient; a single malicious packet (e.g., causing a buffer overflow) or network probe is often just one component or precursor of a larger attack campaign, not an isolated incident.

Subsequent datasets have aimed to address the limitations of KDD'99. CAIDA offers anonymized backbone network traces, but their lack of labels limits their use for supervised learning [14]. UNSW-NB15 features more contemporary network scenarios, though generated artificially using a proprietary tool; studies suggest its attributes yield more realistic model accuracies compared to KDD'99 [15]. ISCX 2012 provides seven days of activity (2 million records), with 2% labeled as attacks like brute-force SSH, infiltration, HTTP DoS, and DDoS [16]. CIC-IDS2017 offers five days of simulated traffic and endpoint data (including network flows, memory dumps and system calls collected via host agents) for several common protocols and attacks [17]. Specialized datasets include ADFA-LD and ADFA-WD for evaluating Linux and Windows HIDS [8], and Bot-IoT, which concentrates on botnet activity within Internet of Things (IoT) environments [18].

Publicly available PCAP dumps, which are complete packet captures collected at some point on the network, might be used instead of a benchmark dataset. However, the usefulness of packet content in modern networks can be limited except in specific cases such as malware analysis. Even so, malware increasingly utilizes encryption and conceals attacks inside unsuspecting protocols such as Domain Name Service (DNS). Intrusion detection systems (IDSs) cannot read encrypted packet content and therefore cannot match encrypted traffic to known signatures. Workarounds include breaking the encrypted connection at the IDS and re-encrypting it using its own certificate. This process introduces vulnerabilities as the network devices become reliant on a single certificate authority for their encrypted communication. In addition, privacy concerns are introduced since the IDS can now see encrypted data. Modern DCS and IDSs consequently tend to rely on statistical network flow features to identify malicious activity instead of analyzing packet content.

2.2. Data Collection Systems

Security Information and Event Management Systems (SIEMs) are foundational tools for aggregating, normalizing, correlating, and storing security-related data from diverse sources across an organization's IT infrastructure. Their correlation capabilities are often enhanced by mapping detected events to frameworks like MITRE ATT&CK [19], which catalogs adversary tactics, techniques, and procedures based on real-world observations, thereby providing context for alerts. While often managing network flow data and various system logs, their effectiveness hinges on the breadth and depth of the data they ingest. Sources can include firewalls, intrusion detection / prevention systems (IDS / IPS) like Suricata [20], antivirus systems, routers, servers, databases, and endpoint security solutions. For instance, platforms like Wazuh [21] utilize agents installed on monitored systems to collect detailed host-level information, including operating system and application logs, configuration changes, and file system modifications, forwarding this securely to a central

server for analysis. However, without such dedicated agents or specific integrations, SIEMs might gather limited information about the network devices and endpoints themselves. Network flow data, which summarize traffic patterns, offer a network-level view and can be generated either directly by network equipment or by dedicated probes. Tools such as Zeek [22] provide another crucial layer by passively analyzing network traffic to produce rich metadata logs detailing protocol-specific activities (HTTP, DNS, SSL, etc.), invaluable for forensic investigations, though typically without active blocking capabilities.

Modern switches and routers often utilize specialized Application-Specific Integrated Circuits (ASICs) for high-speed packet forwarding. Generating flow data directly on these devices, as illustrated in Figure 1, consumes valuable CPU and RAM resources. Caching flow records requires memory proportional to network throughput, and the CPU must actively process packets to create flows and manage their expiration. Network hardware possesses finite processing capacity and can become overloaded in high-throughput environments, especially since these resources are often shared with other critical functions such as firewalling, Quality of Service (QoS) enforcement, and VPN termination. This resource contention can negatively impact overall network performance.

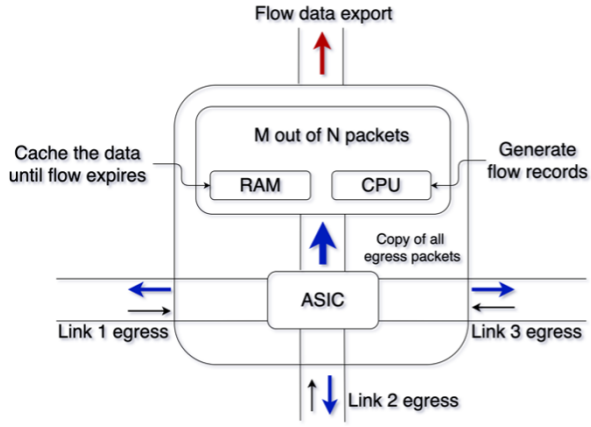


Figure 1. Switch processing of network traffic.

Several standards exist for exporting flow data. NetFlow v5, developed by Cisco, defines a basic flow record using a five-tuple: source and destination IP address, source and destination port, and protocol. To mitigate performance impacts and potential packet drops on busy devices, flow generation can employ sampling techniques. For example, a NetFlow exporter might be configured to sample only one of many packets, creating flow records just for the sampled traffic. Alternatively, sFlow is a standard specifically designed for packet sampling. Both sampling approaches introduce a trade-off between performance and data fidelity. The reduced accuracy can be problematic for detecting stealthy, low-and-slow attacks that generate minimal network traffic over extended periods.

More advanced standards include NetFlow v9, which is a proprietary Cisco protocol that offers richer flow information, and Juniper's similar and also proprietary jFlow. The Internet Engineering Task Force (IETF) developed the IP Flow Information Export (IPFIX) protocol as an open standard comparable to NetFlow v9. Both NetFlow and IPFIX generate unidirectional flow records; accurately accounting for bidirectional traffic often requires careful configuration (e.g., collecting flows on ingress, egress, or both) and potentially deduplication logic at the collector. Although the NetFlow standard originated with Cisco, it is now implemented by many major network vendors. However, support is less common in white-box switches and open-source solutions such as Open vSwitch. Generating flow records using dedicated hardware probes offers greater compatibility across diverse net-

work equipment and ensures data uniformity. It also provides the flexibility to enrich flow records with host data or other contextual information within a single integrated process.

In typical network environments, SIEMs operate alongside various data collection systems. Their primary role involves managing vendor-specific formats and correlating events across disparate sources to facilitate threat detection, often operationalized using frameworks like MITRE ATT&CK to understand adversary behavior. The data aggregated and analyzed by these systems are also crucial for informing the broader risk management processes and compliance requirements outlined in standards such as ISO/IEC 27001 [23], which provides a systematic approach to managing an organization's overall information security posture. Similarly, approaches such as SCOUT highlight the importance of multilayered security, particularly for critical infrastructure, by addressing both cyber and physical attack vectors while incorporating post-incident recovery strategies to balance protection with practical considerations such as cost and privacy [24]. As mentioned, this often includes endpoint data (e.g., from Wazuh agents), network metadata (e.g., from Zeek sensors), and threat alerts (e.g., from Suricata IDS/IPS). InDepth complements these systems by providing detailed and context-enriched flow information while simultaneously offloading the resource-intensive task of flow generation from network infrastructure devices, freeing them for core switching and routing tasks.

Data collection strategies vary significantly. Network monitoring can occur outside the firewall (feeding traditional IDS/IPS like Suricata or network analyzers like Zeek), directly on the firewall (leveraging integrated security features), or on individual hosts via software agents (typical for Host-based IDS like Wazuh). To our knowledge, no existing system simultaneously collects both network flow data and agentless host information, enriching records with VLAN-specific context, without requiring software installation on monitored endpoints—a gap InDepth aims to fill.

Centralized data collection, using one or more aggregation points, presents challenges. Depending on the network topology and location of the collection point, certain traffic flows can generate duplicate records, while others can be completely missed. While some modern routers, e.g., newer Cisco models, can generate deduplicated flow records, they often retain the original, non-deduplicated records, increasing storage requirements.

Furthermore, traditional single-threaded packet capture libraries such as libpcap and PFRING are prone to significant packet loss at high data rates [25]. While CPU core counts have increased, single-core clock speeds face practical limits, leading to diminishing returns in performance scaling. Recognizing this bottleneck, modern network security tools increasingly adopt parallel processing; for example, Suricata employs a multi-threaded architecture to handle high traffic volumes efficiently, similar in principle to multi-threaded capture solutions like DiCAP [26]. This issue of packet loss is particularly acute in centralized collection architectures, such as monitoring at the network backbone, which handle large volumes of traffic and require substantial CPU, memory, and bandwidth resources. In contrast, a distributed data collection system operating at the subnet level can significantly reduce the load on individual collection points, potentially allowing the use of lower-cost commodity hardware or even low-power IoT devices instead of high-end servers.

We propose InDepth as a system that can overcome these limitations. The approach utilizes a multi-threaded, distributed, and node-based architecture that combines passive network flow generation with active host data collection to provide comprehensive and context-aware network visibility.

3. InDepth System Architecture and Cyber Range

InDepth employs a hierarchical system architecture with distinct tiers of components that work together to provide comprehensive network monitoring. At the lowest level, multiple sensor nodes are strategically deployed across different network segments (VLANs), forming the data collection tier. Each sensor node independently monitors its assigned VLAN by receiving mirrored packets from switch ingress ports. Above this collection tier sits the aggregation tier, consisting of a core node that serves as the central integration point for all distributed data. Sensor nodes connect to the core node through encrypted channels, creating a secure data transport layer between the tiers. This hierarchical structure maintains the separation of concerns while ensuring comprehensive coverage. Functionally, the system takes a multi-stage approach. Sensor nodes passively capture network packets through port mirroring, then transform these packets into network flows using parallel processing threads. Simultaneously, separate threads within each sensor node actively probe hosts to gather device characteristics. These two data streams (passive network flows and active host information) are enriched with contextual data before being transmitted to the core node via secure WireGuard VPN channels. The core node, running Elasticsearch, indexes and stores this aggregated data, making it available for analysis and correlation studies. This architecture distributes the computational load while maintaining centralized analysis capabilities, allowing the system to scale effectively across networks of varying sizes and complexities without sacrificing visibility into critical network activities, particularly those that might indicate security threats like lateral movement or data exfiltration attempts.

3.1. Motivation

Generating network flow data with a dedicated external device requires access to raw network packets. Two primary methods provide this access: Network Test Access Points (TAPs) and port mirroring configured on capable switches. TAPs offer the advantage of capturing all traffic on the wire, including physical layer errors that port mirroring might discard, without consuming switch processing resources. However, deploying TAPs on every relevant switch port can be prohibitively expensive. Port mirroring, in contrast, leverages existing network hardware by configuring specific ingress or egress ports to duplicate traffic. This makes it a more cost-effective and feasible solution in many environments. For InDepth, we employ port mirroring on switch ingress ports. To mitigate the risk of packet drops under heavy load, a known potential issue with port mirroring, sensor nodes use multi-threaded data processing to allow computational resources to be scaled for network segments experiencing high throughput.

Port mirroring exists in two main forms: simple and encapsulated. Simple port mirroring creates an identical copy of the traffic observed on the mirrored ports and enjoys broad support across network hardware, including Software-Defined Networking (SDN) environments. Encapsulated port mirroring, such as Remote Switched Port Analyzer (RSPAN) or Encapsulated RSPAN (ERSPAN), wraps the mirrored traffic in specific protocols (e.g., GRE) and can forward it to a remote IP address, potentially outside the local subnet. Although this facilitates centralized data collection, the support for encapsulated mirroring varies among manufacturers. InDepth utilizes simple port mirroring, integrating it with our distributed sensor node architecture to aggregate the collected data centrally at the core node.

Collecting data directly at the subnet level using a distributed approach offers significant advantages compared to centralized collection, particularly at the network backbone. Distributed sensors capture data-link layer (Layer 2) information, including crucial host details like MAC addresses and VLAN IDs, which are often lost in traffic traversing routers

towards a central point. This localized visibility enables the detection of early-stage attack activities, such as internal vulnerability scanning and lateral movement within a subnet. Centralized systems, monitoring traffic further upstream, might only observe the final stage of an attack, such as data exfiltration to an external command and control (C2) server. Furthermore, active host probing, used to gather device characteristics, becomes more reliable and efficient when target hosts are only a single network hop away from the sensor. While the network performance impact of the active probing is negligible, we acknowledge that battery-powered devices might experience additional wake-ups to respond; however, modern mobile operating systems are often optimized to preserve battery life in noisy network environments, which can sometimes delay probe responses. The subnet-centric strategy of combining network activity monitoring, host characteristic gathering, and contextual enrichment effectively distributes the computational load across multiple sensor nodes. This reduces the processing and memory demands on any single point in the system and conserves network bandwidth by transmitting compact metadata records instead of full packet captures.

Modern network traffic patterns increasingly involve communication with the Internet and cloud-based services rather than solely internal resources. Centralized data collection systems located at the network edge or backbone are well positioned to monitor and mitigate threats originating from or targeting the external network, as they observe all traffic entering and leaving the local environment. However, these systems are challenged when processing the extremely high data volumes typical at the backbone, potentially leading to data loss or incomplete analysis. Critically, they often lack visibility into intra-subnet activity. Important phases of sophisticated attacks, such as lateral movement between compromised hosts within the same network segment, may never traverse the backbone and would thus remain undetected by purely centralized monitoring.

Contemporary network infrastructures increasingly rely on software-based implementations, utilizing technologies such as Software-Defined Networking (SDN) and virtualization platforms (e.g., VMware ESXi). These technologies enable network administrators to dynamically allocate compute, memory and bandwidth resources to network functions and virtual servers on demand, often without service interruption. Consequently, modern and future networks demand a data collection framework that operates effectively in both physical hardware and virtualized software environments, scaling seamlessly to accommodate fluctuating network throughput and evolving infrastructure paradigms.

3.2. Data Generation and Storage

As illustrated in Figure 2, each sensor node receives mirrored packets from all ingress ports of its designated VLAN switch. The simple port mirroring produces duplicates of packets seen (barring certain manufacturer-specific error messages) and is compatible with modern switch hardware and virtual networking devices. Crucially, this avoids encapsulation, eliminating the need for additional processing at the sensor. Multi-threading allows the conversion of network packets into network flows to scale with the number of CPU cores available. A separate thread actively probes hosts to gather device data, which are buffered. The flow records are enriched with this and other contextual data. Completed data records are transmitted to the core node for the network segment through a secure WireGuard VPN channel. Each core node aggregates data from all sensor nodes in its VLAN. Core nodes can communicate with each other using a secure tunnel.

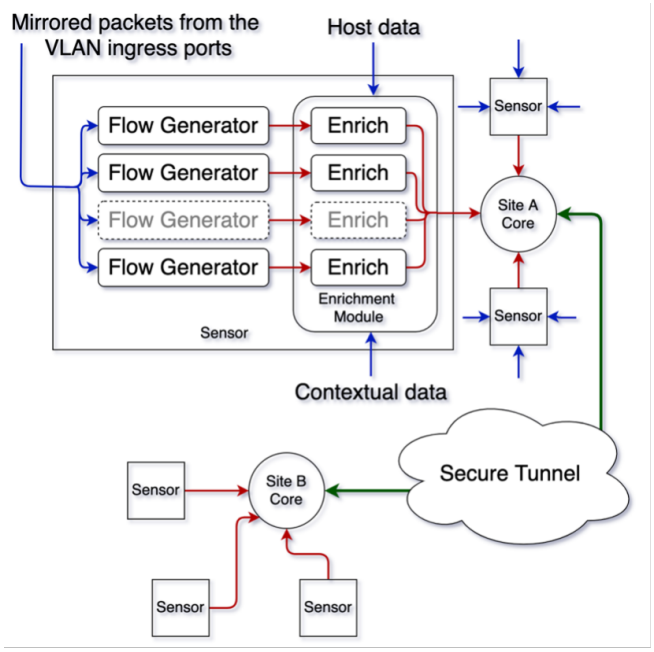


Figure 2. InDepth network flow generation and enrichment.

Our implementation favors open-source software where possible, supplemented by readily available industry-standard tools. For example, we use Argus [27] for flow data generation, nmap [28] for active host probing, Elasticsearch [29] for storage and indexing, and WireGuard [30] to securely transmit sensor node data to the core. Although the virtual network was constructed using Cisco VIRL [31], SDN tools such as Open vSwitch [32] or ONOS [33] could be substituted for a completely open-source stack. This modularity underscores that InDepth is a flexible framework: alternative tools can be used, such as NetFlow v9 instead of Argus for flow generation or ZMap [34] instead of nmap for probing, depending on specific needs. The host devices run a mixture of proprietary and open-source operating systems with various services, as discussed further in Section 3.4.

3.3. Modern Network Stack

Modern networks employ multiple security solutions operating at different layers of the network stack, including network firewalls, web application firewalls (WAF), next-generation firewalls (NGFW), antivirus software, IDS/IPS systems, network situational awareness tools, SIEM platforms, VPNs, and various specialized security tools for wireless, email, and endpoints. Each serves a specific purpose within the security architecture: classic firewalls enforce rule-based connection policies at the backbone router; WAFs function as reverse proxies protecting web applications by examining HTTP/S connections; NGFWs act as forward proxies implementing user-based policies through URL filtering and malware scanning; IDS/IPS systems detect and respond to known attack signatures across multiple protocols (Layers 2–7); situational awareness systems like InSight2 [35] visualize Layer 3 traffic in real-time; and SIEMs aggregate log data from diverse sources throughout the network.

Rather than replacing any of these technologies, InDepth complements the security ecosystem. Operating primarily at Layer 2, InDepth focuses on comprehensive interaction capture and archival by collecting flow data and host information enriched with contextual details. This distributed collection approach provides unique visibility into subnet-level activities that backbone-focused systems might miss, making it particularly valuable for detecting early attack stages such as vulnerability scanning and lateral movement.

3.4. InDepth Cyber Range

The InDepth cyber range simulates a modern network environment for security research, featuring various types of devices and applications. With reference to Figure 3, its hierarchical topology comprises four primary VLANs, namely, Clearnet, Management, Site-to-Site, and DMZ, interconnected via a managed Layer 3 switch, plus a dedicated VLAN for the InDepth system itself. This structure serves as a flexible reference architecture that can be adapted to various network configurations.

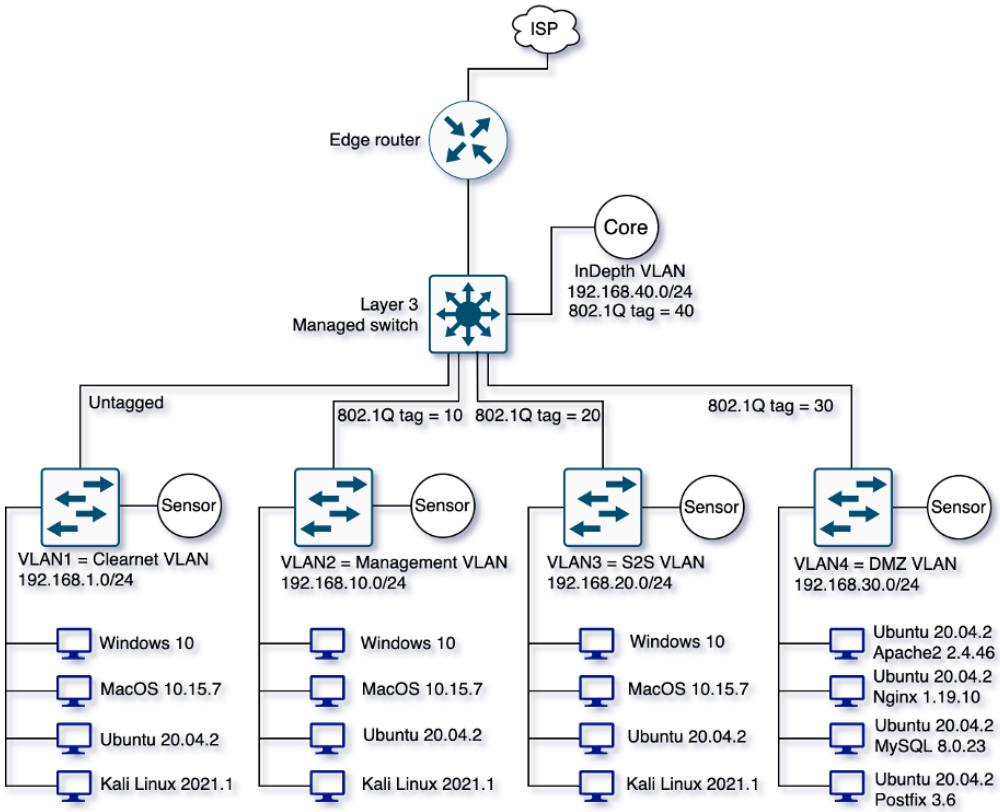


Figure 3. InDepth cyber range.

Traffic from the VLANs converges at the Layer 3 switch, which functions as the default gateway. An edge router then directs traffic to a simulated Internet Service Provider (ISP). The IEEE 802.1Q standard [36] facilitates VLAN tagging. For simplicity in this demonstration, each subnet utilizes a /24 CIDR block, providing 254 usable IP addresses. However, the architecture can readily scale to larger networks through smaller subnet masks (for example, /16, /8) or supernetting. To enable data collection, ingress traffic for each VLAN is mirrored to a dedicated port connected to its corresponding InDepth sensor node. In high-throughput scenarios, Link Aggregation Control Protocol (LACP) can be used to bundle ports, increasing the available bandwidth to the sensor.

The specific roles and configurations of the VLANs are as follows:

- Clearnet (192.168.1.0/24, untagged): Contains standard user devices without special VPN or firewall rules.
- Management (192.168.10.0/24, tag 10): Includes devices used by management personnel, isolated from other subnets via firewall policies.
- Site-to-Site (192.168.20.0/24, tag 20): Establishes an encrypted, isolated connection to a simulated off-site network.
- DMZ (192.168.30.0/24, tag 30): Hosts critical services such as web, email, and database servers in a demilitarized zone. Although such services typically use public IP ad-

dresses, this range employs private IPs for demonstration purposes, as it is not connected to the live Internet, thus avoiding potential conflicts. The installed services run their latest versions but can be replaced with older, vulnerable counterparts (e.g., using OWASP tools [37] or Metasploitable [38]) to facilitate specific security testing and threat signature generation.

- InDepth (192.168.40.0/24, tag 40): An isolated network segment dedicated solely to the InDepth core node, which aggregates data collected from all sensor nodes across the cyber range.

The Clearnet, Management, and Site-to-Site VLANs all host devices that run a mix of operating systems, including Windows, macOS, Ubuntu, and Kali Linux. We note that the Kali Linux host is specifically included to serve as a platform for launching simulated attacks within the range. The DMZ VLAN contains only Ubuntu devices.

Figure 4 lists the network features and the host features collected by the passive monitoring and active probing within this environment.

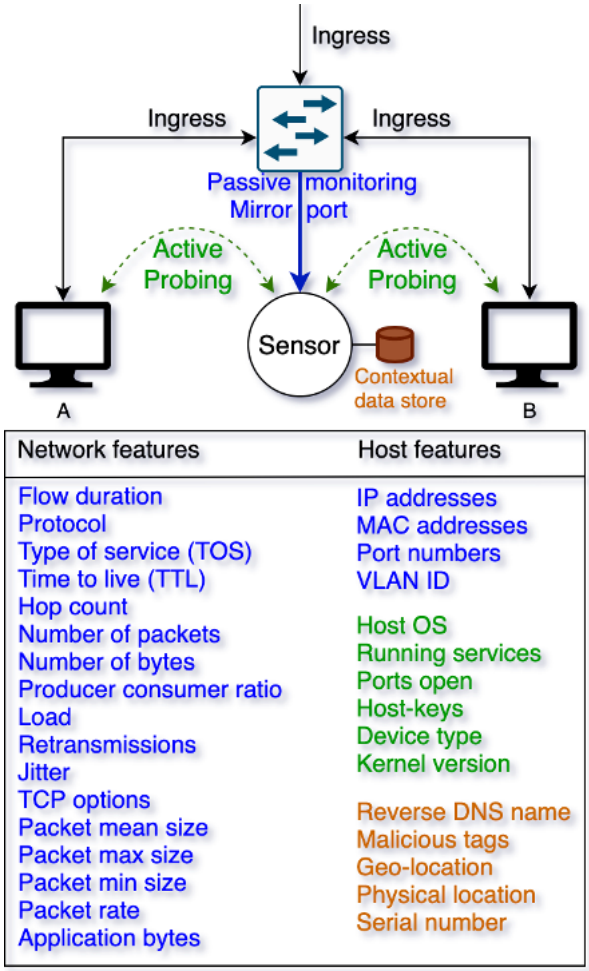


Figure 4. Features collected by InDepth.

4. Experimental Results

4.1. WRCCDC 2020

The WRCCDC 2020 cyber competition was a red team–blue team attack–defend exercise [9]. In previous work, we analyzed the data to identify attacks and compromised machines in the network [35]. When attacking, the red team periodically changed their source IP addresses to prevent being blocked by the blue team's defenses. This practice made it difficult to aggregate their actions and attribute them to the correct source host.

At the gateway router (a Layer 3 switch), only tcpdump data of network packets crossing the gateway were logged, with no additional host information. Contextual host characteristics were not made available for either team. Competition rules gave the blue defending teams an hour to familiarize themselves with the services they were defending and update them to improve resilience against attack. Service versions at the time of compromise are therefore unknown. Most network traffic was collected on the network backbone. The reliance on IP addresses made it challenging to track individual hosts and model their behavior, especially when IP address changes occurred to evade block-listing. In such scenarios, collecting and analyzing all traffic belonging to hosts under scrutiny becomes particularly difficult. The lack of ground truth hindered the completeness of our analysis, despite having some prior information that is typically unavailable in real-world forensic investigations.

The analysis was somewhat simplified by the network topology, which separated red and blue teams into distinct subnets. The red team operated across a large network range of 10.128.0.0/9 when attacking blue team servers. Eight blue teams operated on subnets 10.47.x.0/24, where x represented the team number. A service check engine ran at 10.0.0.111, as shown in Figure 5. However, in real-world environments, attacker and legitimate host IP addresses often intermingle within the same subnet, significantly complicating analysis tasks. This reality underscores the need for more robust and comprehensive data collection mechanisms for continuous network monitoring.

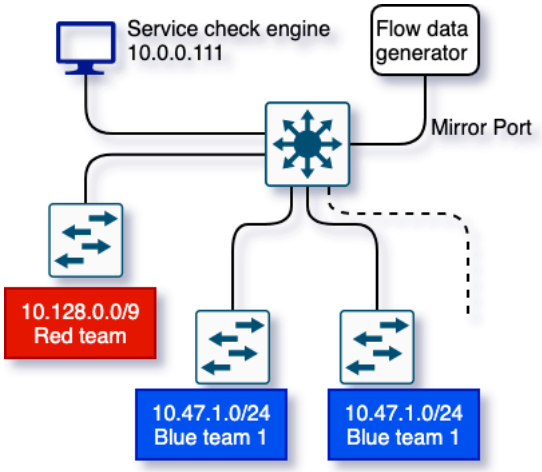


Figure 5. WRCCDC 2020 network topology.

We simulated portions of the WRCCDC 2020 network by creating a red team subnet and a blue team subnet. We first collected network data using the same method as applied in the competition, namely, through the Layer 3 routing device at the backbone connecting the two subnets. Then, we configured InDepth to simultaneously collect both host and network data as shown in Figure 6, where R is the red team host, B1 is a blue team host located in the same subnet as R, and B2 is a blue team host in a different subnet. To maximize data resolution for these experiments, active host probing was configured to run back-to-back, initiating subsequent probes as soon as responses from the previous ones were received.

We performed three experiments that compared the distributed data collection performed by InDepth against standard backbone data collection. It is worth noting that the attack types tested typically represent steps within a larger attack chain rather than standalone events. The results consistently demonstrated that InDepth collected more complete and actionable data across all three scenarios.

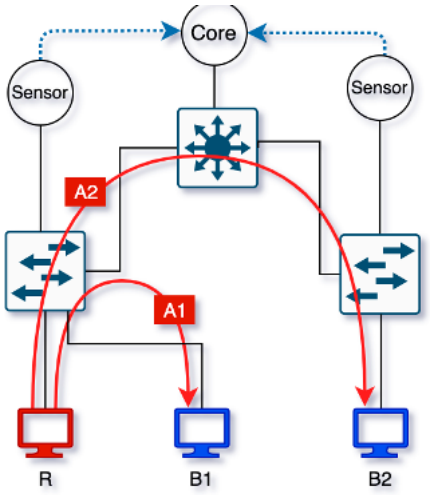


Figure 6. InDepth network topology.

4.2. Experiment 1: Aggregation and Attribution

This experiment evaluated InDepth's ability to aggregate and attribute traffic from a red team device notwithstanding the periodic IP address changes, while providing comprehensive visibility into intra-network activities. Unlike traditional IP spoofing in DoS attacks where responses are ignored, reply packets were required to maintain the attack chains. The red team typically used Metasploit to deliver payloads and establish reverse shells for persistent access, which we successfully replicated in our test environment. When this traffic intermingled with legitimate network activity, traditional backbone monitoring systems struggled to detect the IP address changes, making attacks appear to originate from multiple distinct devices. Using InDepth, on the other hand, we were able to trace all traffic to its true source by leveraging the host-side features and network data collected at the data-link layer.

InDepth's distributed sensor architecture provides comprehensive visibility into intra-network activities by capturing all peer-to-peer communications within LANs, including those that never traverse the network backbone. During security investigations, analysts traditionally face the time-consuming and error-prone task of manually correlating network activity with host attributes, particularly when data-link layer information is unavailable. InDepth addresses this challenge by collecting Layer 2 information (MAC addresses) alongside rich contextual data including operating system details, running services, open ports, host keys, device types, and kernel versions. This comprehensive approach enables real-time tracking of attack hosts throughout the network, regardless of IP address changes or VLAN boundaries.

Traditional security operations rely on reactive attribution techniques such as packet marking, ICMP traceback messaging, hop-by-hop tracing, IPsec authentication logs, and traffic pattern matching. These approaches often depend on specialized network packet crafting, router firmware modifications, manual tracing, configuration access, routing table comparisons, and IPsec security association log analysis. Such processes demand significant time and expertise, typically requiring a team of security specialists and becoming feasible only after detecting large-scale breaches. InDepth's proactive monitoring approach eliminates the need for these reactive measures by maintaining continuous visibility into all network activities, including intra-VLAN communications that traditional systems miss.

4.3. Experiment 2: Lateral Movement

This experiment examined lateral movement, where an attacker used a compromised device to establish connections with other unsuspecting devices across the network. We used InDepth's ability to capture peer-to-peer traffic both within the same VLAN and between different VLANs. In the first scenario (A1 attack), host R connected to host B1 within the same subnet, simulating the delivery of an exploit payload targeting a known vulnerability. In the second scenario (A2 attack), we repeated this procedure with host R targeting host B2 located on a different subnet.

Traditional backbone data collection systems failed to detect A1 attack connections, completely missing evidence of lateral movement within the subnet. In contrast, InDepth successfully identified the A1 attack by correlating network traffic with host characteristics, including the operating system and device type of host R. Both data collection mechanisms detected the A2 attack since it traversed the network backbone.

This scenario highlights a common attack pattern where adversaries first compromise a low-priority device (like B1 in our experiment) to establish an initial foothold before moving laterally through the network. According to CrowdStrike research [39], the average "breakout time", which is the interval between initial compromise and lateral movement, is less than two hours. Detecting and responding to attacks within this critical window is essential to prevent widespread damage.

Although network administrators typically rely on Endpoint Detection and Response (EDR) tools to identify such attacks, these tools often lack comprehensive visibility throughout the network. Meanwhile, InDepth's real-time collection of both network traffic and host information provided the integrated visibility necessary for timely detection and response to the lateral movement attacks.

4.4. Experiment 3: IP Address Spoofing

This experiment examined the detection of spoofed source IP addresses by leveraging InDepth's ability to capture network flows at both source and destination points. When denial-of-service (DoS) attacks are executed, attackers typically randomize source addresses because they do not require return packets, making attribution difficult through conventional means.

Buffer overflow attacks represent one of the most prevalent DoS techniques in real-world scenarios. In these attacks, servers become overwhelmed with traffic volumes exceeding their processing capacity, rendering systems unavailable to legitimate users. To evade firewall blocks, attackers frequently randomize source IP addresses, attempting to blend their malicious traffic with legitimate communications.

Flood attacks commonly manifest in two primary forms. ICMP flood attacks transmit spoofed packets that ping multiple network devices simultaneously. When network misconfigurations exist, these packets can be amplified throughout the infrastructure, effectively disabling network services. Alternatively, SYN floods initiate but never complete TCP handshakes, exhausting available server ports and preventing legitimate connection establishments.

Although large-scale DoS attacks often originate from numerous compromised devices on the Internet, similar attacks can also emerge from within internal networks. InDepth's comprehensive monitoring approach provided a significant advantage in these scenarios. By capturing connections at both the origin and destination points while tracking the characteristics of the physical device, InDepth successfully identified attack flows even when IP addresses were spoofed, maintaining attribution capabilities that traditional collection systems lack.

4.5. Discussion

Our experimental evaluation demonstrates the significant advantages of InDepth's comprehensive data collection approach over traditional backbone-based collection systems. The experimental setup, which recreated portions of the WRCCDC 2020 network environment, allowed us to systematically compare both approaches under controlled conditions. By implementing one red team subnet and one blue team subnet with varying connection patterns, we established a realistic testbed to evaluate security monitoring capabilities.

The experiments, which focused on aggregation and attribution, lateral movement, and IP address spoofing, represent common challenges in network security monitoring that traditional systems struggle to address effectively. In the first experiment, InDepth successfully tracked attacking hosts despite IP address changes by collecting Layer 2 information alongside contextual host data. This capability dramatically reduces the time-consuming manual correlation typically required in security investigations. The second experiment revealed a critical blind spot in backbone collection systems, which completely missed lateral movement within the same subnet (A1 attack), while InDepth maintained full visibility across both intra-subnet and inter-subnet movements. Finally, the third experiment demonstrated InDepth's ability to maintain attribution capabilities even when facing IP spoofing attempts during DoS attacks by capturing flows at both source and destination points.

These results highlight how InDepth's distributed collection architecture provides security analysts with more complete and actionable data compared to traditional approaches. By capturing both network traffic and host information in real-time, InDepth enables more efficient detection and analysis of sophisticated attack patterns. This advantage becomes particularly crucial when considering the increasingly short timeframes within which security teams must identify and respond to threats before attackers can establish persistent access or move laterally through networks. The comprehensive data collection approach of InDepth represents a significant advancement in network monitoring capabilities, addressing many of the limitations that have historically hampered effective security analysis and incident response.

5. Conclusions

We have presented InDepth, a data collection system capable of collecting network and host information from real-world networks, be they physical or virtual. The system consists of a distributed network of sensor nodes that enrich the Layer 2 data by contextual information and a core node where the data are aggregated. InDepth supports total interaction capture from all hosts in the network including peer-to-peer communication at each subnet. The collected data can be used to develop and train machine learning models, but can also be used by security analysts for training purposes as well as for conducting forensic studies following a breach incident. We also introduced a model cyber range which can be used to generate new datasets that can replace the outdated ones still widely used for analysis and algorithm development. We used this cyber range to compare InDepth against standard backbone data collection capabilities for three attack scenarios. We found the data collected by InDepth to be more complete and provide more useful information. Future work includes the deployment of InDepth on select networks to build said replacement benchmark datasets.

Author Contributions: Conceptualization, A.K.; methodology, A.K. and J.G.; project administration, J.G.; software architecture, A.K.; supervision, J.G.; validation, A.K. and J.G.; software implementation, A.K.; writing—original draft, A.K.; writing—review and editing, A.K. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the National Science Foundation under Grant No. IRNC-1450959.

Data Availability Statement: Data available in a publicly accessible repository. The original data presented in the study are openly available in WRCCDC 2020 archive [9].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. 1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 30 March 2025).
2. KDD Cup 1999 Data. Available online: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 30 March 2025).
3. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Dataset. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
4. McHugh, J. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Trans. Inf. Syst. Secur.* **2000**, *3*, 262–294. [CrossRef]
5. Mahoney, M.V.; Chan, P.K. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. In *Recent Advances in Intrusion Detection*; Giovanni, V., Christopher, K., Erland, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 220–237.
6. Maciá-Fernández, G.; Camacho, J.; Magán-Carrión, R.; García-Teodoro, P.; Therón, R. UGR'16: A New Dataset for the Evaluation of Cyclostationarity-based Network IDSs. *Comput. Secur.* **2018**, *73*, 411–424. [CrossRef]
7. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
8. Creech, G.; Hu, J. Generation of a New IDS Test Dataset: Time to Retire the KDD Collection. In Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 4487–4492. [CrossRef]
9. WRCCDC Public Archive. Available online: https://archive.wrccdc.org/pcaps/2020/ (accessed on 30 March 2025).
10. Kodituwakku, A.; Gregor, J. InMesh: A Zero-Configuration Agentless Endpoint Detection and Response System. *Electronics* **2025**, *14*, 1292. [CrossRef]
11. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
12. KDDCUP99 Dataset. Available online: https://www.tensorflow.org/datasets/catalog/kddcup99 (accessed on 30 March 2025).
13. Kayacik, H.G.; Zincir-Heywood, N.; Heywood, M. Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99. In Proceedings of the Third Annual Conference on Privacy, Security and Trust, St. Andrews, NB, Canada, 12–14 October 2005. Available online: https://www.semanticscholar.org/paper/Selecting-Features-for-Intrusion-Detection%3A-A-on-99-Kayacik-Zincir-Heywood/60e28c7da56eb61dd8ddb710a6f079ef02668014 (accessed on 30 March 2025).
14. Yavanoglu, O.; Aydos, M. A Review on Cyber Security Datasets for Machine Learning Algorithms. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2186–2193. [CrossRef]
15. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Dataset for Network Intrusion Detection Systems. In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
16. Intrusion Detection Evaluation Dataset (ISCXIDS2012). Available online: https://www.unb.ca/cic/datasets/ids.html (accessed on 30 March 2025).
17. Panigrahi, R.; Borah, S. A Detailed Analysis of CIC-IDS2017 Dataset for Designing Intrusion Detection Systems. *Int. J. Eng. Technol.* **2018**, *7*, 479–482.
18. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
19. MITRE ATT&CK. Available online: https://attack.mitre.org/ (accessed on 30 April 2025).
20. Suricata. Available online: https://github.com/OISF/suricata (accessed on 30 April 2025).
21. Wazuh. Available online: https://github.com/wazuh/wazuh (accessed on 30 April 2025).
22. Zeek. Available online: https://github.com/zeek/zeek (accessed on 30 April 2025).
23. *ISOIEC 27001:2022*; Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements. ISO: Geneva, Switzerland, 2022. Available online: https://www.iso.org/standard/27001 (accessed on 30 April 2025).

24. Cantelli-Forti, A.; Capria, A.; Saverino, A.L.; Berizzi, F.; Adami, D.; Callegari, C. Critical infrastructure protection system design based on SCOUT multitech seCurity system for intercOnnected space control groUnd staTions. *Int. J. Crit. Infrastruct. Prot.* **2021**, *32*, 100407. [CrossRef]
25. Wu, H.; Liu, Y.; Ni, S.; Cheng, G.; Hu, X. LossDetection: Real-Time Packet Loss Monitoring System for Sampled Traffic Data. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 30–45. [CrossRef]
26. Morariu, C.; Stiller, B. DiCAP: Distributed Packet Capturing Architecture for High-speed Network Links. In Proceedings of the 2008 33rd IEEE Conference on Local Computer Networks (LCN), Montreal, QC, Canada, 14–17 October 2008; pp. 168–175. [CrossRef]
27. Bullard, C. Argus Software. Available online: https://github.com/openargus/argus (accessed on 30 March 2025)
28. Nmap: The Network Mapper. Available online: https://nmap.org/ (accessed on 30 March 2025).
29. Elasticsearch. Available online: https://github.com/elastic/elasticsearch (accessed on 30 March 2025).
30. Donenfeld, J.A. WireGuard VPN Tunnel. Available online: https://www.wireguard.com/ (accessed on 30 March 2025).
31. VIRL. Available online: https://learningnetwork.cisco.com/s/virl (accessed on 30 March 2025).
32. Linux Foundation. Open vSwitch. Available online: https://www.openvswitch.org/ (accessed on 30 March 2025).
33. Open Networking Foundation. Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions. Available online: https://opennetworking.org/onos/ (accessed on 30 March 2025).
34. Durumeric, Z.; Wustrow, E.; Halderman, J.A. ZMap: Fast Internet-Wide Scanning and its Security Applications. Available online: https://github.com/zmap/zmap (accessed on 30 March 2025).
35. Kodituwakku, H.A.D.E.; Keller, A.; Gregor, J. InSight2: A Modular Visual Analysis Platform for Network Situational Awareness in Large-Scale Networks. *Electronics* **2020**, *9*, 1747. [CrossRef]
36. IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks. In IEEE Std 802.1Q-2022 (Revision of IEEE Std 802.1Q-2018) **2022**. [CrossRef]
37. Sagar, D.; Kukreja, S.; Brahma, J.; Tyagi, S.; Jain, P. Studying Open Source Vulnerability Scanners for Vulnerabilities in Web Applications. *IIOAB J.* **2018**, *9*, 43–49.
38. Metasploitable 2 Exploitability Guide Documentation. Available online: https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/ (accessed on 30 March 2025).
39. Crowdstrike Global Threat Report. Available online: https://www.crowdstrike.com/resources/reports/2020-crowdstrike-global-threat-report (accessed on 30 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.