Holoscope: Open and Lightweight Telescope & Honeypot Platform

ACCEPTED FROM OPEN CALL

Holoscope: Open and Lightweight Telescope & Honeypot Platform

Andrea Sordello, Marco Mellia, Idilio Drago, Rodolfo Valentim, Francesco Musumeci, Massimo Tornatore
Federico Cerutti, Martino Trevisan, Alessio Botta, and Willen B. Coelho

Abstract

The complexity and scale of Internet attacks call for distributed, cooperative observatories capable of monitoring malicious traffic across diverse networks. Holoscope is an open, lightweight, and cloud-native platform designed to simplify the deployment and management of telescope (passive) and honeypot (active) sensors. Built upon K3s and WireGuard, Holoscope offers secure connectivity, automated sensor onboarding, and resilient operation even in resource-constrained environments. Through modular design and Infrastructure-as-Code principles, it supports dynamic sensor orchestration, automated recovery, and data processing. We build, deploy, and operate Holoscope across multiple institutions and cloud networks in Europe and Brazil, enabling unified visibility into large-scale attack phenomena while maintaining ease of integration and security compliance.

Introduction

Security monitoring infrastructures play a central role in identifying emerging cyber threats and characterising malicious Internet activity. Among these, network telescopes (darknets) monitor unused, routable IP address space that hosts no services. Since no legitimate traffic should target these addresses, the unsolicited traffic they receive typically consists of scanning campaigns targeting exposed hosts and services, unveiling potential malicious behaviour such as worm propagation and botnet reconnaissance [1]. In contrast, honeypots are decoy systems that emulate vulnerable services, enabling active interaction with attackers and observation of exploitation attempts, command execution, lateral movement, and post-compromise activities.

Darknets and honeypots are often deployed within a single network domain, making the observed traffic strongly dependent on factors such as network prefix size, autonomous system (AS), and geographic location, as malicious activities are often localised on specific address ranges [2].

Achieving a broader and more comprehensive view of malicious behaviour requires combining observations from multiple heterogeneous networks [3–6]. However, building such collaborative infrastructure is challenging due to the need for secure data sharing, automation, scalability, and cross-organisational management.

We introduce Holoscope, a distributed and scalable cybersecurity observatory developed within the PROTECT-IT/SERICS [7] project. Holoscope simplifies the deployment and management of Internet-exposed sensors across geographically distributed networks owned by different organisations. It enables a configurable network of sensors that continuously collect, process, and share unsolicited traffic from diverse vantage points, producing high-value data for modelling attack patterns and studying large-scale adversarial behaviour. We provide an overview of Holoscope in Fig. 1.

At the time of writing, Holoscope is running across 10 organisations — including universities, service providers, and cloud infrastructures — in Europe, the US, and Brazil. It runs on Unix-based physical or virtual systems, ensuring broad deployability. The platform is publicly available at https://github.com/SmartData-Polito/Holoscope. The traffic dataset is available under a non-disclosure agreement due to the sensitive nature of the traffic traces.

The remainder of the article is organised as follows: we review related work on distributed telescopes and honeypot networks; we present Holoscope's design, operation, and modules; we describe its real-world deployment and observed traffic insights; and discuss the lessons learned.

Related Work

Distributed infrastructures for collecting unsolicited Internet traffic have evolved along two main directions:
1. Large-scale darknet platforms [1] and
2. Coordinated honeypot systems [8].

We consider recent solutions that are not deprecated, support replication through distributed sensors, and operate in real-world deployments rather than simulations or single-organisation testbeds. A representative list is provided in Table 1.

At the national scale, distributed darknet initiatives include the US-based Merit National Dis-

ISSN: 0163-6804
Digital Object Identifier: 10.1109/MCOM.001.2500784

Andrea Sordello, Marco Mellia, Idilio Drago, Rodolfo Valentim, Francesco Musumeci, Massimo Tornatore, Federico Cerutti, Martino Trevisan, Alessio Botta, and Willen B. Coelho

tributed Network Telescope (NDNT) [9] and the Greek EWIS platform [10], both of which federate sensors across multiple administrative domains. Unlike Holoscope, EWIS depends on preconfigured hardware distributed to participants, limiting scalability and flexibility. NDNT focuses on federating existing telescope sensors and IP address space, rather than offering a reproducible open-source deployment framework. Cloud-based distributed telescope designs have also been explored [6, 11], demonstrating the feasibility of deploying darknet sensors in cloud environments. Holoscope likewise supports cloud-hosted sensors, enabling elastic deployment, geographic diversity, and multi-provider scalability.

The honeypot ecosystem has likewise evolved toward distributed architectures. Recent versions of the widely used T-Pot [12] support distributed deployments, and other platforms, such as SweetsPot [5], enable the orchestration of multiple honeypots across geographically and administratively distributed environments. In contrast to Holoscope, these platforms primarily focus on honeypot deployment, with limited emphasis on sensor configuration and orchestration mechanisms. Other solutions, such as GCA AIDE [13], are built around proprietary honeypot technologies.

A smaller set of platforms integrates both darknet and honeypot monitoring. One example is CAIDA's iVoyager [14], a novel project that combines a central experimentation node with lightweight distributed sensors. Compared to Holoscope, which prioritises large-scale darknet data collection and analysis, iVoyager places stronger emphasis on offering a centralised research infrastructure.

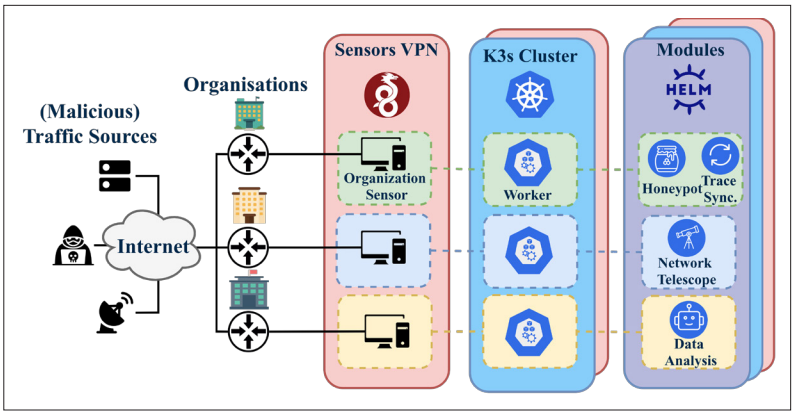


FIGURE 1. Holoscope architecture. Each organisation provides a sensor exposed to the Internet. Holoscope activates modules to collect and analyse malicious traffic.

Platform	Platform of ...	3rd-Party Sensor	Sensor Config.	Experiment Set
SweetsPot [5]	Honeypot	✓	×	Custom
GCA AIDE [13]	Honeypot	✓	×	Proprietary
T-POT [12]	Honeypot	✓	×	T-POT
Merit NDNT [9]	Darknet	–	–	–
EWIS [10]	Darknet	✓	✓	–
Bortoluzzi et al. [11]	Darknet	×	✓	–
iVoyager [14]	Both	✓	✓	Custom
Holoscope	**Both**	✓	✓	Custom

Legend: ✓ Supported, × Not supported, – Not applicable/Not provided

TABLE 1. Overview of existing platforms.

Holoscope

Holoscope must satisfy two primary requirements:

1. Supporting operation across distributed and heterogeneous resource-constrained sensors
2. Providing a flexible and responsive execution environment for applications.

Holoscope Design

We now describe the design of Holoscope's architectural layers, motivating their purpose and explaining how they interact with one another.

Secure Connectivity Layer: The foundation of Holoscope is a secure layer that interconnects all participating sensors, located across the networks of third-party organisations. Holoscope relies on a Virtual Private Network (VPN) that provides confidentiality and integrity of communications while normalising networking across heterogeneous environments. This abstraction mitigates issues related to Network Address Translation (NAT), firewalls, and dynamic IP addressing. The VPN topology follows a point-to-point model between each sensor organisation and the central server(s). This design prevents direct sensor-to-sensor visibility and limits lateral movement in case of compromise, strengthening the overall security posture. We implement this layer using WireGuard, a lightweight modern VPN solution.

Orchestration Layer: To meet the requirements of flexibility and responsiveness in application management, Holoscope adopts cloud-native technologies. Docker containers provide an effective mechanism to package and distribute applications across sensors by encapsulating each application together with its dependencies into lightweight, isolated environments. This approach ensures portability across heterogeneous devices and prevents conflicts arising from differing system configurations.

However, distributing containers across multiple geographically dispersed nodes requires automated coordination. For this reason, the orchestration layer leverages a container orchestrator to establish a distributed cluster on top of the secure network. The orchestrator is responsible for scheduling workloads, maintaining their desired state, handling failures, and enforcing isolation and security policies through an off-the-shelf solution. By automating deployment, scaling, and recovery, the orchestration layer enables Holoscope to dynamically adapt to changes in application requirements and variations in node availability.

We implement the orchestration layer using K3s, a lightweight Kubernetes distribution. The adoption of K3s provides several advantages, including:

- Efficient operation in resource-constrained environments
- Application availability and self-healing through Kubernetes primitives
- Redundancy via replicated *etcd* nodes, improving platform resiliency

Module Layer: The top layer of Holoscope comprises modular applications deployed by the container orchestrator. We refer to these applications as *modules* because each provides a self-contained application that can be independently executed without affecting the rest of the platform. A module includes not only the applica-

To automate and systematically control these configuration tasks, we adopt Ansible, an open-source automation engine that manages infrastructure through reusable playbooks, i.e., declarative configuration code.

tion container but also the associated Kubernetes resources, e.g., services and storage components, required for its correct operation. Example modules include darknet, honeypots, and traffic analysis tools. We present the complete list later.

Holoscope Operation

We now describe how Holoscope manages its sensors, focusing on the processes for onboarding new ones, performing their initial configuration, maintaining and updating deployed sensors, and tracking organisation-specific requirements, such as network interface configurations and IP addressing.

To automate and systematically control these configuration tasks, we adopt Ansible, an open-source automation engine that manages infrastructure through reusable playbooks, i.e., declarative configuration code. This enforces the Infrastructure-as-Code (IaC) paradigm as a core principle of Holoscope, ensuring reproducibility, consistency, auditability, and scalability across heterogeneous and geographically distributed environments, while reducing manual intervention and configuration drift.

New Sensor Onboarding: A fundamental operational task in Holoscope is the onboarding of new sensors. To enable this process, the sensor must be reachable over the Internet via SSH, which allows the automated execution of the Ansible playbook responsible for this task. This playbook

- Installs the required software components, including Docker, K3s, and WireGuard;
- Configures the additional virtual interface for the VPN connection that will be used for the sensor's management; and
- Registers the sensor within the K3s cluster.

By automating the entire onboarding process, the Ansible-based approach significantly reduces manual intervention by the sensor owner and ensures a consistent, reproducible deployment process.

Adding Modules to Holoscope: Holoscope offers a private container image registry that we use to deploy modules. The choice to offer a private registry avoids depending on a third-party registry, while keeping our modules secure. To add a new module to Holoscope, it is sufficient to provide a *Dockerfile*, which can be built and uploaded to the private registry.

Deploying Modules on Sensors: A key aspect of Holoscope is the heterogeneity of its sensors. Each sensor has its own configuration, e.g., network interface card (NIC) names, IP address ranges as well as specific policies. To handle this heterogeneity, we employ Helm, a Kubernetes package manager that allows creating parametrised, reusable flexible deployments. By using Helm, we define module templates and later customise them with sensor-specific details, thereby extending the Infrastructure-as-Code paradigm also to module deployment.

Modules Offered by Holoscope

We now present the set of core applications available in Holoscope. We refer to them as modules, as each operates independently and runs as a self-contained component.

Darknet — Passive Telescope: The Darknet module activates a passive sensor over an unused IP address range owned by the organisation. It supports three deployment configurations:

1. The organisation statically routes the entire darknet prefix from the ingress router to the sensor's production interface. This requires a one-time modification of the routing policy.
2. The organisation assigns the full darknet address space directly to the sensor's NIC. This is the simplest setup: unsolicited traffic reaches the sensor as regular traffic, and no responses are generated. Cloud-based darknet deployments commonly adopt this model, treating darknet IPs as standard routable addresses [6, 11].
3. A custom module performs transparent Layer-2 redirection by replying to ARP requests from ingress routers, associating darknet IPs with the sensor's MAC address and forwarding traffic accordingly. This approach has been adopted in prior darknet deployments [10, 15].

In all configurations, the Darknet module coordinates with the Network Enforcement module to install appropriate iptables rules that block any outgoing traffic from the darknet address range.

Honeypot — Active Responder: Holoscope supports any honeypot that can be containerised. Deploying containerised honeypots within a cluster introduces two main challenges:

1. Exposing the pod hosting the honeypot transparently, and
2. Preventing any lateral movement originating from the pod itself, as it runs vulnerable code.

We address the first challenge through the Network Enforcement Module which manages traffic steering and port exposure. To preserve honeypot integrity and minimise the risk of compromise, we rely on Kubernetes' built-in isolation mechanisms, ensuring each pod operates with the minimum required privileges. Since this is a static approach, we additionally integrate Falco, an active monitoring system that tracks runtime behaviour and detects signs of compromise.

Network Enforcement: Although Kubernetes provides native networking abstractions, such as ClusterIP and NodePort services, these mechanisms are insufficient for darknet and honeypot deployments. Such scenarios require fine-grained traffic steering and enforcement policies beyond standard Kubernetes services.

For instance, we must block any outgoing packets generated from darknet IP addresses to preserve their passive nature. Moreover, the exposure of a honeypot using a sensor port inevitably requires using the Kubernetes *hostNetwork* option. However, this creates a security risk, since the pod would have unrestricted access to the sensor's root network namespace.

Motivated by these limitations, we introduce the novel Network Enforcement Module to obtain a more fine-grained traffic control mechanism.

Its task is to handle unsolicited traffic arriving at the sensor's NIC and forward it to the appropriate module (e.g., honeypot or darknet), making them appear as if they are running as real services. This redirection is done by inserting proper rules inside the sensor's iptables and executing the module as a privileged DaemonSet. At runtime, the module generates and updates the sensor's iptables based on the modules currently running on the sensor, enforcing different exposure rules dynamically. Thanks to this steering at the kernel level, we limit the overhead.

Sensor	Organization	Country	Network Telescope	Honeypot Allowed?	Daily /24 statistics	
					# Packets	# Senders
A_1, A_2	**Politecnico di Torino**	ITA	/23	✓	4.1M	45k
B_1, B_2	Consortium GARR	ITA	/23	–	3.4M	44k
C	Politecnico di Milano	ITA	/23	✓	1.3M	37k
D	Università di Trieste	ITA	/22	–	1.9M	42k
E	Università di Brescia	ITA	/24	✓	18M	101k
F_1	UFES	BRA	/25	✓	2.7M	42k
F_2	IFES	BRA	/25	✓		
G	RNP	BRA	/19	–	6.3M	42k
H	Microsoft Azure	East US	/24	✓	4.7M	48k
I	Univ. di Napoli Federico II	ITA	–	✓	–	–
Holoscope		**Global**	**Available 11 520 IPs**	**Generally Allowed**		

TABLE 2. Organizations contributing to Holoscope.

The use of this module provides additional advantages, including rate limiting — crucial to prevent the sensor from inadvertently participating in Distributed Denial of Service (DDoS) attacks — and the ability to expose a honeypot instance across a range of ports or IP addresses without requiring multiple deployments. For example, Holoscope can route traffic from attackers to different backend services or steer packets targeting selected address ranges to basic Layer-4 responders that complete the TCP three-way handshake to capture initial payloads [3].

The module prevents interference with the native iptables chains managed by the K3s Container Network Interface (CNI) by using additional custom iptables chains.

Traffic Collector: This module captures all traffic directed to the monitored address space of each sensor, while excluding management traffic through properly configured capture filters. It uses tcpdump to save packet traces on the sensor's disk.

Log Sync: The module retrieves raw traffic traces and other logs from sensors and stores them in a centralised storage using Rsync, a data transferring tool. Organisations can limit or disable this feature, e.g., in cases where organisation-specific data retention policies apply.

Data Analysis: Finally, Holoscope supports modules dedicated to analysing the traffic and logs collected by the sensors. Because these tasks may impose significant computational overhead, we typically execute them on the most powerful sensor available, subject to the hosting organisations' policies. The analyses range from simple aggregated metric computation to more advanced techniques leveraging machine learning and artificial intelligence (ML/AI).

Deployment of Holoscope

We now describe our deployment and provide an overview of 3 months of collected traces.

Participating Organisations

We deploy Holoscope through collaboration with several Italian and Brazilian organisations, as shown in Table 2. All organisations provide an IP range within their address space to install a darknet. Some support honeypot deployment and workload execution, e.g., data processing, directly within their infrastructure.

Each organisation contributes subnet ranges of varying sizes and sensor capabilities (ranging from 4 GB RAM and 4 vCPU to 32 GB RAM and 16 vCPU). To enable a fair comparison, in our analysis, we consider only a /24 subnet per sensor. Sensors F_1 and F_2, which correspond to two contiguous /25 networks located in two different campuses, are treated jointly as a single /24 sensor, and denoted as *F*. The sensors joined Holoscope at different times. For instance, we activated sensor *H*, hosted in Microsoft Azure East US, in August 2025. A single /24 darknet typically observes between 1.3 million and 6.3 million packets per day from approximately 42 thousand unique external senders.

Resource Overhead

We evaluate Holoscope's computational resource requirements by quantifying the platform overhead relative to standalone experiment execution. To avoid impacting the production deployment described earlier, we conduct a benchmark by deploying Holoscope on dedicated VMs (2 vCPUs and 2 GB of RAM each).

Holoscope introduces an overhead of approximately 300 MB of RAM and 3% to 10% CPU usage per sensor. The sensor acting as the K3s master node has higher resource requirements, consuming between 0.6 and 1 GB of RAM and around 25% CPU, due to additional components running on it, e.g., the K3s *etcd* database.

Holoscope requires only a few GB of disk space, while the volume of downloaded data during the installation is on the order of MB. The management traffic exchanged, over the VPN, by sensors consists predominantly of K3s/VPN management messages.

Holoscope Traffic Profile

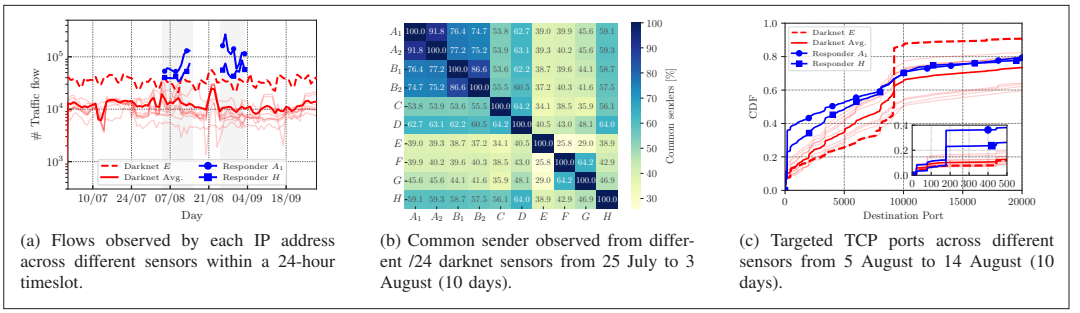
We analyse traffic collected by Holoscope sensors from July 2025 to October 2025. During this period, some sensors hosted at the same time both active and passive experiments, we refer to them as "Responder *X*" and "Darknet *X*". We examine traffic in terms of:

1. Temporal evolution
2. Sender similarity,
3. Targeted destination ports.

Traffic Temporal Profile: Figure 2a illustrates the temporal evolution of flows received by each

We evaluate Holoscope's computational resource requirements by quantifying the platform overhead relative to standalone experiment execution.

Leveraging cloud-native technologies and the Infrastructure-as-Code (IaC) paradigm, it provides secure, automated, and resilient orchestration across heterogeneous networks.



(a) Flows observed by each IP address across different sensors within a 24-hour timeslot.

(b) Common sender observed from different /24 darknet sensors from 25 July to 3 August (10 days).

(c) Targeted TCP ports across different sensors from 5 August to 14 August (10 days).

FIGURE 2. Traffic characteristics observed across distributed darknet sensors.

IP address across sensors. We define a flow by its standard 5-tuple. For each sensor, we monitor a /24 subnet with only the darknet module enabled (red curves; the solid line denotes the average). On average, each IP observes approximately 10^4 flows per day, with fluctuations reflecting variations in external sender activity.

The darknet at sensor *E* emerges as a clear outlier, receiving roughly three to four times more flows than other sensors with a distinct periodic pattern. Manual inspection reveals significantly higher backscatter traffic — TCP SYN/ACK packets from potential DDoS attack victims where adversaries spoof random source IP addresses, mainly from source ports 53, 80, and 443. Since the *E* address space resembles private ranges such as 192.168.0.0/16, senders may inadvertently or intentionally select a similar address.

During the period shown in Fig. 2a, we activated twice a basic L4 responder in a /28 subnet within sensors A_1 and *H* in August (grey period). They observe an increase in traffic flows, peaking above 10^5 per day. While similarly configured, the sensors received different traffic volumes, with A_1 receiving about twice the traffic (note y-log scale). This confirms that active responders change traffic profiles, and Holoscope enables observations from multiple, geographically diverse vantage points.

Common Senders on Different Sensors: We now focus on differences among senders contributing to unsolicited traffic across sensors. Different sensors receive traffic from distinct sender sets; however, overlaps occur. Figure 2b quantifies the ratio of common senders observed by each /24 darknet sensor pair from 25 July to 3 August. To reduce noise, we focus only on highly active senders, defined as senders generating at least 500 packets over the 10-day observation period. This subset represents approximately 5% of all observed senders.

Italian sensors A_1, A_2, B_1, B_2, *C*, and *D* share over 50% of senders, likely because their networks belong to the same Autonomous System (AS), making their address ranges similar to senders. Overlap increases to 86%–91% for sensors on contiguous address blocks (*A* and *B* cases).

Brazilian sensors *F* and *G* share around 64% of senders, reflecting similar regional visibility within their AS. Interestingly, sensor *H* (Microsoft Azure East US) shows a stronger affinity with Italian sensors (around 58% overlap) than Brazilian ones (43%), suggesting cloud-based sensors attract sender populations more akin to those targeting academic or enterprise networks in Europe rather than Latin America. Sensor *E*, which records the highest unique senders, shares fewer senders with any sensor, confirming its outlier status.

These results demonstrate the added value of a distributed observatory: only by aggregating observations from multiple vantage points can we discern spatial diversity and sensor-specific visibility biases.

Destination Port Across Modules

Figure 2c shows the cumulative distribution function of TCP SYN traffic per destination port targeted by senders across sensors. Darknet sensors exhibit similar profiles, with traffic concentrated on consistent ports: 22 (SSH), 23 (Telnet), and 2000, commonly associated with brute-force and IoT exploitation. Responder sensors (A_1 and *H*) show dissimilar port distributions to darknets and each other: they experience noticeable activity on port 179 (BGP — see inset). Notably, A_1 receives 30% of packets on port 179, compared to only 10% for *H*. *H* sees more traffic on ports 1024–10000, consistent with its steeper CDF slope. This confirms that active response modules increase engagement, but different locations yield different results.

Sensor *E* is a notable exception: its darknet receives over 40% of TCP SYN traffic on port 9200, typically used by Elasticsearch. This distinct pattern aligns with its anomalous flow volume, suggesting specific automated DDoS campaigns may disproportionately target that address space.

Overall, these findings reinforce the importance of heterogeneous, geographically distributed vantage points: they reveal not only volumetric differences but also distinct behavioural signatures across attackers, networks, and deployment contexts.

Lessons Learned and Conclusion

We presented Holoscope, an open, lightweight cybersecurity platform. Leveraging cloud-native technologies and the Infrastructure-as-Code (IaC) paradigm, it provides secure, automated, and resilient orchestration across heterogeneous networks. Our initial multi-sensor deployment demonstrated its scalability and its ability to reveal traffic insights unattainable from a single vantage point. The deployment experience highlighted two critical aspects of operating a distributed infrastructure. First, ensuring cluster resiliency is essential, as sensor failures can disrupt experiments and lead to resource waste. Second, comprehensive experiment automation is necessary to minimise human error and prevent misconfigurations that could compromise the quality of the collected data.

Two main directions will guide our future work. First, we plan to expand the platform by introducing a novel set of experiments, such as those involving DNS and certificate announcements, while leveraging this unique and flexible infrastructure to devel-

op and foster AI-driven data analytics capabilities. Second, we aim to disseminate actionable artefacts to the cybersecurity community, including datasets, periodic reports, blocklist insights, and integrations with existing threat reporting platforms.

Acknowledgement

This work was supported by the SERICS (PE00000014) and the ACRE (2022EP2L7H) projects under the MUR National Recovery and Resilience Plan funded by the European Union — NextGenerationEU. This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

References

[1] F. Bortoluzzi, B. Irwin, and C. M. Westphall, "Cloud Telescope: An Ephemeral, Distributed, and Cloud-Native Architecture for Collecting Internet Background Radiation," *IEEE Access*, vol. 13, 2025, pp. 45,682–45,714.
[2] P. Richter and A. Berger, "Scanning the Scanners: Sensing the Internet From A Massively Distributed Network Telescope," *ACM IMC*, 2019, pp. 144–57.
[3] F. Soro *et al.*, "Are Darknets all the Same? On Darknet Visibility for Security Monitoring," *IEEE LANMAN*, 2019, pp. 1–6.
[4] D. Wagner *et al.*, "How to operate a meta-telescope in your spare time," ser. IMC '23. ACM, 2023, p. 328–343.
[5] T. Angeli *et al.*, "Demo: SweetsPot: A Distributed Honeypot Federation Platform," *IEEE LCN*, 2025, pp. 1–4.
[6] E. Pauley, P. Barford, and P. McDaniel, "Dscope: A Cloud-Native Internet Telescope," *Proc. 32th USENIX Security Symp.*, 2023.
[7] Fondazione SERICS — SEcurity and RIghts In the CyberSpace, accessed on 18 Feb 2026; https://serics.eu/.
[8] J. Franco *et al.*, "A Survey of Honeypots and Honeynets for Internet of Things, Industrial internet of Things, and Cyber-Physical Systems," *IEEE Comm. Surveys & Tutorials*, vol. 23, no. 4, 2021, pp. 2351–83.
[9] National Distributed Network Telescope — Merit, accessed on 18 Nov. 2025; https://www.merit.edu/research/national-distributed-network-telescope/.
[10] P. Chatziadam, I. G. Askoxylakis, and A. Fragkiadakis, "A Network Telescope for Early Warning Intrusion Detection," *Springer HAS*, 2014, pp. 11–22.
[11] F. Bortoluzzi *et al.*, "Cloud Telescope: A Distributed Architecture for Capturing Internet Background Radiation," *2023 IEEE CLOUDNET*, 2023, pp. 77–85.
[12] T-Pot — The All In One Multi Honeypot Platform, accessed on 23 Feb. 2026; https://github.com/telekom-security/tpotce.
[13] AIDE — Global Cyber Alliance, accessed on 20 Nov. 2025; https://gcaaide.org/.
[14] Internet Voyager for Gathering Cyber Threat Intelligence (iVoyager), accessed on 18 Feb. 2026; https://www.caida.org/funding/cns-ivoyager/.
[15] M. Bailey *et al.*, "Practical Darknet Measurement," *IEEE CISS*, 2006, pp. 1496–1501.

Biographies

Andrea Sordello (andrea.sordello@polito.it) He is a PhD student at Politecnico di Torino within the SmartData@Polito research center. His research focuses on network traffic collection and analysis.

Marco Mellia [F'21] (marco.mellia@polito.it) He is a full professor at the Politecnico di Torino. His research interests are in Internet monitoring, cybersecurity, and AI applied to different sectors.

Idilio Drago (idilio.drago@unito.it) He is an associate professor at the Università di Torino. His research interests include network security, machine learning, and Internet measurements.

Rodolfo Valentim (rodolfo.viera@polito.it) is a research assistant specialising in Machine Learning and Cybersecurity at Politecnico di Torino, working on AI solutions for traffic analysis.

Francesco Musumeci [SM'23] (francesco.musumeci@polimi.it) He is an associate professor at Politecnico di Milano, Italy. His research interests are in the fields of ML-aided networking, cybersecurity, converged space-ground networks and disaster resilience.

Massimo Tornatore [F'22] (massimo.tornatore@polimi.it) He is a full professor at Politecnico di Milano, Italy. His research interests include performance evaluation and design of communication networks and machine learning applications for network management.

Federico Cerutti [SM'23] (federico.cerutti@unibs.it) He is a full professor at the Università di Brescia, Italy and visiting professor at Imperial College London, UK. His research interests are in the areas of cyber threat intelligence and security of AI.

Martino Trevisan (martino.trevisan@dia.units.it) He is an associate professor at the Università di Trieste, Italy. His research interests include network measurements, big data, and cybersecurity.

Alessio Botta (a.botta@unina.it) He is an associate professor at the Universià di Napoli Federico II. His reseach interests include traffic measurements with applications to cybersecurity, and usage of AI.

Willen B. Coelho (willen@ifes.edu.br) He is currently a PhD student at UFES and IT analyst at the IFES, both in Brazil.