NotLine: a Non-Intrusive Automated Platform to Build a Digital Twin

F. Baiardi*, V. Sammartino*, S. Ruggieri*
*Dipartimento di Informatica, Università di Pisa, Italia
Email: {fabrizio.baiardi, salvatore.ruggieri}@unipi.it
vincenzo.sammartino@phd.unipi.it

***Abstract*—Digital twin technology is revolutionizing cybersecurity by providing real-time, data-driven replicas of ICT infrastructures without impacting live production systems. We present NotLine, a *non-intrusive, fully automated pipeline* platform that builds and updates digital twins through the continuous passive collection of multiprotocol network traffic metadata. NotLine filters and normalizes the data to remove noise and then correlates events to generate a dynamic topology graph. This non-intrusive approach enhances network monitoring and mitigates the risks and overhead associated with active scanning. It also offers superior scalability and enables continuous threat hunting, risk assessment, and accelerated remediation.**

The resulting digital twin extends traditional static inventories with real-time vulnerability mapping through CVE lookups. It also supports AI-driven adversary simulations based on Monte Carlo methods. Furthermore, we explore how integrating non-intrusive host-level telemetry, threat intelligence feeds, and reinforcement learning can evolve the digital twin into a self-optimizing cybersecurity guardian.

Experiments in production environments demonstrate that passive monitoring over extended periods, spanning multiple days, is essential for accurately capturing daily and weekly usage patterns across diverse protocol families. A quantitative analysis is presented that establishes benchmarks for digital twin fidelity in networked environments.

***Index Terms*—Digital Twin, Passive Network Monitoring, Non-Intrusive Discovery, Automated Pipeline, Network Topology, Adversary Simulation, Vulnerability Management, Risk Assessment.**

I. Introduction

The exponential growth in the complexity and scale of modern information and communication technology (ICT) infrastructures underscores the need for cybersecurity approaches that can adapt to dynamic threat landscapes. In particular, the digital twin paradigm, a virtual representation that evolves in lockstep with its physical counterpart, offers a continuous, data-driven insight into the state of the network and its exposure to vulnerabilities [9, 10].

This paper introduces NotLine, a fully automated, **Not-intrusive pipeLine** platform to build and maintain an accurate security twin: a specialized digital twin of ICT environments focused on cybersecurity. NotLine builds on passive network monitoring tools (e.g., ntopng [8]) to collect heterogeneous protocol metadata, which it then filters, normalizes, and correlates by continuously updating a graph that describes devices, services, and their interactions. With respect to monitoring, NotLine integrates real-time vulnerability feeds and AI-driven adversary simulation through Monte Carlo methods [12, 13] to support proactive threat hunt, comprehensive risk assessment, and validation of mitigation strategies, all without requiring active scanning or disrupting production systems.

The remainder of this paper is structured as follows. First, Sect. II analyzes the gaps in current network discovery paradigms and highlights our main contributions. Next, Sect. III reviews the literature on digital twins and passive monitoring in cybersecurity, setting the stage for our approach. Sect. IV presents the NotLine platform and its end-to-end pipeline for building a security twin. Then Sect. V delves into the technical details of each component of the pipeline and the architecture that underpins them. We describe a first experimental setup, quantitative findings, and discuss fidelity requirements in Sect. VI. Finally, Sect. VIII summarizes our conclusions and suggests avenues for future work.

II. Gap Analysis and Research Contribution

The evolving complexity of modern networks poses major challenges in ensuring visibility and security posture. This section analyzes the limitations of current approaches, outlines our research contribution to address these gaps, and examines related work in the field.

A. Limitations of Current Discovery Paradigms

Network discovery methodologies traditionally fall into two separate categories, active and passive, each with inherent strengths and limitations that affect their effectiveness in contemporary environments.

Active discovery provides detailed asset information through direct interaction, but its critical drawback is the generation of additional network traffic, which can trigger security alerts whenever intrusion detection systems interpret scanning activities as potential threats [6]. More concerning is their impact on sensitive operational technology environments, where probes could reduce both the communication bandwidth and the computational one in industrial control systems and lead to production downtime. Organizations recognize these risks and typically restrict active scanning to narrow maintenance windows. This may result in significant blind spots in situational awareness between scheduled scans [15].

Conversely, passive network discovery offers a non-intrusive alternative by monitoring traffic without injecting additional packets. Despite this advantage, passive methods suffer from their constraints. The most significant is incomplete visibility:

Devices that communicate infrequently or remain idle for extended periods may completely escape detection [5]. Although all devices are eventually detected, there is a substantial delay, as precise identification requires capturing a significant volume of traffic [11]. Network architectures further complicate this approach, as effective passive monitoring is highly dependent on strategic sensor placement and access to traffic flows, which may be impractical in some network segments or distributed environments.

These limitations highlight a critical gap in network discovery capabilities: Organizations lack methodologies that provide comprehensive real-time asset visibility without introducing operational risks or requiring excessive infrastructure changes.

B. Our Contribution: Continuous Passive Discovery

Modern networks are constantly evolving: devices continuously connect and disconnect, their configurations evolve, and new vulnerabilities are continuously discovered. Approaches relying on periodic active scanning or manual inventory processes introduce unacceptable latency, operational risk, and potential service disruption.

NotLine is a *non-intrusive* pipeline platform that addresses these challenges as it extends passive network information collection and monitoring into a comprehensive asset discovery that enriches any asset with security attributes. NotLine continuously ingests and processes high-volume protocol feeds, including ARP, mDNS, SSDP, ICMP, and DHCP transactions, and applies advanced normalization and correlation algorithms to produce a coherent and continuously updated network topology graph. The full automation of NotLine offers distinct advantages:

- *Enhanced Operational Safety:* By avoiding active probing, NotLine also avoids any risk and overhead associated with active scanners. Hence, it can be applied even to the most sensitive network environments.
- *Seamless Scalability:* The decentralized nature of passive collection allows NotLine to scale across large-scale or geographically distributed networks without requiring scan windows or changes to the infrastructure.
- *Real-Time Synchronization:* Unlike point-in-time scans, NotLine continuous monitoring captures every network event, from new node deployments to security patch deployments, in near real time, ensuring that the asset inventory is continuously updated. As a further advantage, there is no human in the loop.

NotLine creates and updates a *security twin* that supports advanced cybersecurity capabilities (see e.g., [2, 4, 17]). Anytime it discovers new assets or new vulnerabilities are disclosed, NotLine immediately updates the security twin and it runs Monte Carlo adversary simulations on the updated twin to uncover potential attack paths enabled by the updated network conditions and it quantifies the resulting risk. Lastly, it suggests remediation to apply before intrusions occur [13, 16].

Perhaps most significantly, NotLine establishes a self-reinforcing feedback loop where observed anomalies, remediation actions, and topology changes continuously refine the detection models and correlation thresholds. In this framework, the digital twin is no longer a passive monitoring solution and it becomes a security twin – an adaptive, self-optimizing cybersecurity tool that grows more accurate and responsive over time. According to the classification in [19], a security is designated as Level 4 (Prescriptive) as, without human intervention it can predict future intrusions and provide recommendations according to current and anticipated conditions.

III. Related Work

Digital twins have been extensively studied in engineering and manufacturing contexts, where virtual replicas support design, monitoring, and predictive maintenance [7, 9]. In cybersecurity, early explorations highlighted the potential of digital twins for risk assessment and vulnerability management [3, 18], offering significant advantages compared to traditional static inventories that quickly become obsolete in dynamic environments.

Network monitoring tools such as ntopng have enabled a scalable collection of multi-protocol metadata, significantly simplifying rich traffic analysis and anomaly detection across complex infrastructures [8]. Our work aligns with recent research trends that have focused on enriching these diverse telemetry streams and merging them into comprehensive graph models for threat modelling and attack path discovery[23].

Adversary simulation techniques, including AI-driven and Monte Carlo methods, have been successfully applied to static network models to uncover potential attack vectors and assess defensive strategies under various conditions [12, 13]. Our framework advances these approaches by integrating simulation capabilities with a continuously updated digital twin, delivering accurate, real-time insights into an organization's evolving security posture.

Although current studies have discussed individual components such as monitoring, graph-based modeling, and simulation approaches in isolation, the proposed framework unifies these elements into a comprehensive automated real-time pipeline that is designed for large-scale operational environments. This integration addresses the fundamental limitations of available discovery paradigms while enabling advanced security capabilities previously impractical in production settings. The NotLine platform maintains an updated view of network topology and asset relationships while minimizing the operational risks and visibility gaps that arise in traditional methods.

IV. NotLine: A General Overview

Constructing the security twin involves a dynamic process that integrates live network monitoring with sophisticated data analysis. At its core, NotLine integrates a set of tools into a pipeline designed to capture data from a computer network and to continuously update the twin to accurately reflect the current state of an evolving system. Next, we enumerate the main features of the NotLine platform.

a) Data Acquisition and Real-Time Monitoring: The first step of the pipeline applies ntopng [8] to collect an uninterrupted stream of network metadata from passive monitoring of protocols such as ARP, SSDP, mDNS, ICMP, IPv6 multicast, and DHCP. This capture layer returns the raw input to build the twin without injecting traffic or disrupting production operations.

b) Data Processing and Correlation: Then NotLine filters and normalizes captured metadata. This step removes duplicates and irrelevant noise, aligns the timestamps, and standardizes the fields. The correlation mechanism connects IP addresses to MAC addresses through ARP, ties DHCP leases to specific device configurations, and links service announcements (such as mDNS/SSDP) with host identities. This results in a cohesive set of events prepared for modeling.

c) Vulnerability Information Management: A dedicated vulnerability module in NotLine contains CVE records, vendor advisories, and internal pen-test findings. This information is updated with a predefined frequency by accessing some vulnerability databases. The access returns the vulnerabilities that affect each asset and stores them in a central vulnerability table in the twin. Each entry in the table points to all affected nodes, services, or configuration elements. In turn, each of these points back to the entry. This eliminates duplication and ensures consistent updates: when the severity or the exploitability of a vulnerability changes, or a patch is available, the table is updated once, and all pointers automatically reflect the new information.

d) Continuous Update and Synchronization: As new nodes are discovered, or existing nodes are inactive for some time, NotLine updates the topology graph and creates or removes nodes (devices), edges (communications or dependencies). In this way, every time a node connects or disconnects from the network or its configuration changes, NotLine updates the current twin as soon as the node interacts with another node. Also, NotLine updates the topology graph when new vulnerabilities are discovered that affect some system nodes.

e) Visualization: The platform renders the security twin —i.e. the evolving graph of the infrastructure— on an interactive dashboard, where users can pan, zoom and query device properties. Color-coded status indicators highlight newly discovered hosts, configuration drifts, or service changes. This visual representation gives security teams an immediate, intuitive understanding of network shape and activity.

f) Continuous Collection: The collection pipeline in NotLine does not adopt scheduled scans or manual polling, and it is always active. Hence, NotLine persistently listens to all supported protocol feeds. This uninterrupted data stream guarantees that short-lived devices or ephemeral service broadcasts are captured and retained for analysis.

g) Anomaly Detection: Built-in heuristics monitor for deviations from established baseline patterns—e.g. spikes in ICMP echo rates, unexpected DHCP lease churn, or unusual multicast group changes. Detected anomalies generate alerts and metadata tags, flagging events for deeper inspection or automated response.

h) Periodic Assessment: At predefined intervals (e.g. weekly, monthly, or quarterly, depending on organizational policy and risk appetite), NotLine runs a risk assessment of the security twin. In this assessment, NotLine recomputes vulnerability assignments, revalidates configuration compliance, and reruns simulations taking into account the latest vulnerabilities that have been discovered. This update to the twin is not due to changes in the system the twin describes, but rather to inferences drawn from offline risk assessment.

i) Analysis and Simulation (Monte Carlo): Beyond period assessment, NotLine includes modular engines that leverage further contextual information, such as the threat model of interest. For example, the AI-driven risk engine runs Monte Carlo adversary simulations that explore potential attack paths as enabled by the current topology and vulnerability assignments, quantifying likelihoods of compromise and highlighting high-risk propagation chains. Results help prioritize mitigation strategies before intrusions occur. Other engines can be defined in a modular way according to the problem of interest.

V. NotLine: A detailed view of the pipeline

The pipeline in NotLine includes modular components for data collection, preprocessing, model construction, and adversary simulation. Figure 1 shows NotLine high-level architecture.

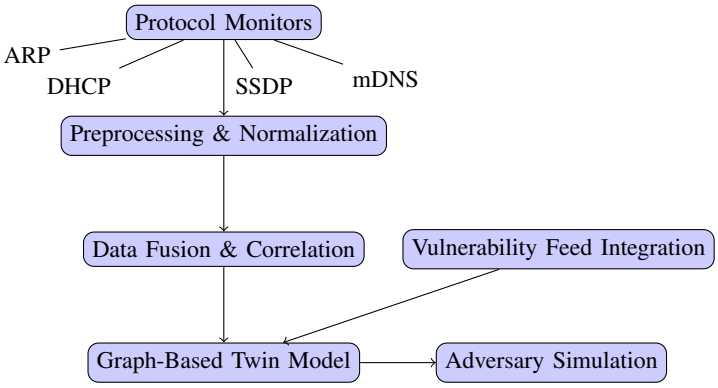

Fig. 1. NotLine Platform Architecture.

A. Protocol Monitoring and OS Inference

Each protocol monitor ingests raw traffic and extracts metadata:

- ARP, mDNS, SSDP, IPv6 Multicast, ICMP, DHCP: capture device presence, service announcements, and communication patterns.
- The inference of the OS uses TTL values, TCP window sizes, and protocol-specific header fingerprints to estimate the operating system without active scanning [1].

B. Analysis of Network Protocols

Understanding the output of the analysis of each protocol is crucial in building and updating an accurate security twin.

These outputs support the discovery of the current infrastructure, providing insight into active devices, communication patterns, and potential vulnerabilities. We detail each key protocol and its contribution to the twin:

- **ARP (Address Resolution Protocol):** ARP is used to map IP addresses to their corresponding MAC addresses within a broadcast domain. When building a digital twin, ARP is fundamental for identifying active devices on the network. By capturing ARP requests and responses, the system can dynamically update the inventory of connected devices, ensuring that the twin accurately represents the physical interconnection network.
- **IPv6 Multicast Groups:** IPv6 utilizes multicast groups to efficiently send data to multiple hosts simultaneously. Monitoring IPv6 multicast traffic reveals active group memberships and service communications. The discovery of the devices that belong to a multicast group helps to map the network topology and to identify clusters of devices that may be functionally or geographically related.
- **SSDP (Simple Service Discovery Protocol):** SSDP is mainly used in UPnP (Universal Plug and Play) environments to discover devices and services within a network. The service announcements it broadcasts highlight the features of each device. SSDP data simplifies the creation of a security twin by automatically identifying active services on a host, providing information about possible vulnerabilities, and offering a more comprehensive view of the network's service landscape.
- **mDNS (Multicast DNS):** mDNS resolves hostnames to local IP addresses and avoids a dedicated DNS server. It is essential for network discovery, as it allows devices to advertise their names and services dynamically. mDNS offers valuable information for a digital twin about the identity and roles of devices, enhancing the accuracy of the network map and facilitating better asset management.
- **ICMP (Internet Control Message Protocol):** ICMP supports the communication of error messages and diagnostic information related to network operations. It helps in detecting unreachable hosts, network congestion, and routing issues. By monitoring ICMP messages, the platform can detect anomalies and network faults, providing an additional layer of detail to the network topology and helping to identify potential vulnerable areas.
- **DHCP (Dynamic Host Configuration Protocol):** DHCP automates the assignment of IP addresses to devices on a network. Analysis of DHCP transactions provides insight into network configuration, including lease times and device behavior patterns. This analysis helps to dynamically track changes in the network environment, ensuring that the digital twin remains up-to-date with current IP allocations and host configurations.

C. Device Discovery Example: Lucias-iMaclocal

We show the information NotLine inserts in the security twin to model the sample device `Lucias-iMaclocal` as a dynamic asset enriched with real-time network monitoring and protocol-specific insights. Beyond basic network-level parameters, the platform can capture extended metadata essential for vulnerability analysis and asset management. Key attributes maintained for each device include:

- **Identification and Network Details:**
  - **Symbolic Name:** Lucias-iMaclocal
  - **MAC Address:** 3C:A6:F6:6E:FA:23
  - **IP Address:** 131.114.X.XX (IPv4)
  - **Local Host Status:** True (indicating internal network presence)
- **Communication and Flow Statistics:**
  - **Flow Peer Analysis:** Discovers active communication between the client (`Lucias-iMaclocal:17500`) and the server (131.114.X.XX:17500) via UDP, utilized by cloud services (e.g., Dropbox).
  - **Traffic Statistics:** Total observed traffic of 2.95 MB with a goodput ratio of 91.2%.
  - **Protocol Breakdown:** Detailed metrics gathered from multiple protocols such as ICMP, UDP, TCP, and mDNS, alongside categorical NDPI statistics (System, Network, Cloud, etc.).
- **Operating System and Software Details:**
  - **OS Identification:** NotLine recognizes the device as an iMac. Extended discovery methods (e.g., mDNS and SSDP interrogation) infer that it is running macOS. NotLine can also record in the security twin the operating system version (e.g., macOS Monterey or Ventura) based on service banners or agent-based reporting.
  - **Software Inventory:** NotLine stores metadata on installed applications and patch levels to correlate it with available security advisories.
- **Activity Timelines and Performance Metrics:**
  - **Timestamps:** First seen on 14/04/2025 and last seen on 16/04/2025, with a total activity period captured.
  - **Throughput Metrics:** Recorded trends in packet per second (pps) and bit per second (bps) provide information on network performance and potential device anomalies.
- **Essential Digital Twin Attributes:** NotLine continuously updates these attributes to reflect the evolving nature of the network environment:
  - Unique identifiers (MAC, IP, and symbolic name)
  - Communication flows and protocol-specific statistics
  - Operating system details and software inventory with version information
  - Historical activity and performance data for trend analysis
  - NDPI-based categorization of network behavior (e.g., Cloud, Streaming)

D. Vulnerability Enrichment

To enable adversary simulation, NotLine enriches the twin with public vulnerabilities through the following steps:

1) **Metadata Extraction:** NotLine extracts detailed data from the device, such as operating system version, software inventory, patch levels, and network behavior from the twin.
2) **Database Query:** NotLine uses device metadata to query centralized vulnerability databases (e.g., the National Vulnerability Database or the Common Vulnerabilities and Exposures Database [14, 20]). The query specifies identifiers like OS version, application banners, and service fingerprints to retrieve matching CVE entries.
3) **Vulnerability Mapping:** When the query returns some matching CVEs, NotLine maps these vulnerabilities to the corresponding device attributes. For each vulnerability, key details are documented:
   - **CVE Identifier:** A unique code identifying the vulnerability.
   - **Severity Score:** Metrics such as CVSS score that indicate potential impact.
   - **Exploitability:** Information on how the vulnerability might be exploited in a network context.
   - **Remediation Status:** Details on available patches and mitigation recommendations.
4) **Alerting and Reporting:** NotLine automatically generates an alert when a device's vulnerability status changes or when new vulnerabilities are disclosed. These alerts support prompt remediation actions.
5) **Feedback Loop:** Post-remediation, updated vulnerability status is fed back into the digital twin, refining future assessments.

An Example: Vulnerabilities for `Lucias-iMaclocal`. Consider the sample device `Lucias-iMaclocal` integrated into the digital twin. The device, identified as an iMac running macOS, is currently running a version older than the fixed releases. During periodic vulnerability assessment (as defined in Section IV), NotLine queries the vulnerability database using the device's operating system details and service banners. The query returns a match with the following details:

- **CVE Identifier:** `CVE-2025-31194`
- **Vendor:** Apple Inc.
- **Published Date:** 2025-03-31
- **Description:** An authentication issue was addressed with improved state management. This vulnerability allows a Shortcut to run with administrative privileges without proper authentication.
- **Affected Products:** macOS versions prior to:
  - macOS Ventura 13.7.5
  - macOS Sequoia 15.4
  - macOS Sonoma 14.7.5
- **Severity:** High (CVSS metrics indicate a significant risk due to the potential for unauthorized privilege escalation.)
- **Remediation:** The vulnerability is fixed by upgrading to one of the secure releases (e.g., macOS Ventura 13.7.5 or later).

Given that `Lucias-iMaclocal` is running a vulnerable version (macOS Monterey), it is linked to the CVE-2025-31194 vulnerability in the security twin.

VI. Experimental Results and Discussion

This section presents our first experimental findings related to the building of a security twin. We examine three critical aspects that determine the effectiveness of NotLine in creating a security twin:

- the relationship between the number of captured messages and accuracy,
- the persistence time to ensure accuracy,
- the node coverage network monitoring achieves.

Our analysis applies both traditional network metrics (duration, throughput, total bytes transferred) and novel metrics designed to capture the nuances of flow data in building a security twin. The analysis neglects the potential loss of packets by the monitoring tool, as this issue is negligible over longer observation intervals, where the following packets can recover the lost information.

A. Network Structure Definition

We first describe the logical and physical interconnections of the monitored network that operates in a University department. In our experiments, NotLine collected data from a single physical interface (`eno1`), over which 372 concurrent flows traversed 77 distinct devices. At the core sits a central monitoring server, which aggregates flow records via ntopng. Around it, multiple client hosts—both IPv4 and IPv6—communicate using TCP (HTTP, SSH) and UDP (mDNS, NetBIOS, Dropbox, SSDP, LLMNR) protocols. The broadcast and multicast domains (224.0.0.251 for mDNS, 239.255.255.250 for SSDP) link the clients in Layer 2, while the server resides behind a forwarding node that funnels all traffic to `eno1`. In our monitoring scenario, we distinguish two main IPv4 subnets: 131.114.x.x and 146.48.x.x. Their traffic characteristics and topology are explored in Section VI-C, dedicated to node coverage.

Key elements:

- **Central server:** Collecting and timestamping every flow record over `eno1`.
- **Client hosts:** Some 77 devices (laptops, desktops, IoT) perform HTTP, SSH, Dropbox syncs, name resolution (mDNS, NetBIOS, LLMNR) and discovery (SSDP).
- **Transport and overlay:** Mixed TCP/UDP, with broadcast/multicast groups spanning the whole Layer-2 segment.
- **Measurement interface:** Single NIC (`eno1`), capturing 0–202 kbit/s aggregated throughput (183 pps) over 42 days of uptime.

Because every flow, regardless of the protocol, shares the same interface, the total number of packets and bytes directly

Protocol	Total Bytes (MB)	Percentage (%)
HTTP	835.8	61.3%
NetBIOS	239.6	17.6%
MDNS	153.2	11.2%
Dropbox	134.0	9.8%
SSH	2.75	0.2%
Total	1365.35	100%

TABLE I
VOLUME OF TRANSFERRED DATA PER PROTOCOL.

determines how many messages NotLine needs to achieve statistical convergence in metrics such as throughput, duration, and byte counts.

B. Accuracy vs Time to Build the Digital Twin

An estimate of the number of messages NotLine requires to build an accurate security twin can optimize resource utilization while preserving accuracy. The flow data analysis has discovered a significant pattern in message volume distribution across different protocols and applications.

The analysis shows that HTTP connections to the monitoring server result in the highest volume data transfers, with 472.41 MB and 310.95 MB for the two top flows, respectively. These connections persisted for approximately 1 day and 20 hours each, suggesting that sustained HTTP messaging is critical for digital twin accuracy.

The analysis reveals that a minimum threshold of approximately 250 messages per monitored network provides sufficient data points to build an initial version of the twin. This threshold was estimated by examining the correlation between message count and the accuracy of the topology graph using different protocols. In summary (see Tables I and II):

- HTTP flows show a high information density, with each message contributing significant modelling data.
- A larger number of messages of Broadcast protocols (MDNS, NetBIOS) (more than 200) have to be captured to establish reliable patterns due to their periodic nature
- Application-specific protocols (Dropbox, SSH) showed intermediate requirements (50-150 messages)

Furthermore, we observed that message diversity due to different protocols significantly improves the accuracy of the twin. The most accurate security twins need data from at least four different protocols. This suggests that *protocol diversity is as important as the raw message count*.

Persistence time, the time interval when NotLine monitors network flows, directly impacts the accuracy of the resulting twin. Our experimental data spans a wide range of persistence times, from less than 1 second to over 42 days, providing a comprehensive basis for analysis. The longest-persisting flows in our dataset include MDNS (42 days, 3:11:16), NetBIOS name service (42 days, 3:11:12), and NetBIOS datagram service (42 days, 3:11:13). However, these long-duration flows are characterized by low throughput (246.4 bit/s for MDNS and 147.2 bit/s for NetBIOS), indicating periodic, sparse communication rather than continuous data exchange.

The minimum persistence time that NotLine requires to build an accurate security twin depends on the network activity patterns:

Protocol	Flows	Percentage
HTTP	50	26 %
mDNS	25	13 %
NetBIOS	38	20 %
Dropbox	42	22 %
SSH	2	1 %
Other	35	18 %
Total	**192**	**100 %**

TABLE II
NUMBER OF FLOWS PER PROTOCOL.

- For continuously active services (HTTP, SSH): A minimum of 24 hours of monitoring is required to capture daily usage patterns.
- For sporadic services (Dropbox, Spotify): At least 5-7 days are necessary to observe representative behavior patterns.
- For infrastructure services (MDNS, NetBIOS, DHCP): At least 3 days are required to capture periodic announcements and renewals.

The accuracy of a twin improves significantly when persistence time exceeds one week, as this captures weekly patterns in network usage. The most comprehensive twins in our experiments were built using at least 5 days of continuous monitoring, capturing both workday and weekend network behaviour patterns.

Notably, the two primary HTTP flows to the monitoring server had durations of approximately 1 day and 20 hours, and each transferred sufficient data (472.41 MB and 310.95 MB) to establish baseline network behaviour, but lacked representation of longer cyclical patterns that emerged in flows with multi-week persistence.

C. Node Coverage Analysis

Node coverage considers the percentage of infrastructure nodes that the twin represents, that is, the percentage of nodes the analyses of the protocols discover. Node coverage determines the accuracy of the twin. The data on input flows offers significant insights into the percentage of node coverage that NotLine can reach.

The target system includes distinct types of nodes:

- Server systems.
- Personal workstations (identified by hostnames like mt-pc-dip, ale-workstation).
- Mobile devices (various Android and iOS phones).
- Infrastructure systems (broadcast addresses, multicast groups).

Based on the flow data, we observed the following.

- Primary subnet coverage: The 131.114.x.x range shows extensive coverage with numerous inter-node communications.
- Secondary subnet coverage: The 146.48.x.x range appears only in specific client-server interactions.
- IPv6 network segment: Limited but present, primarily in discovery and neighbor solicitation protocols.

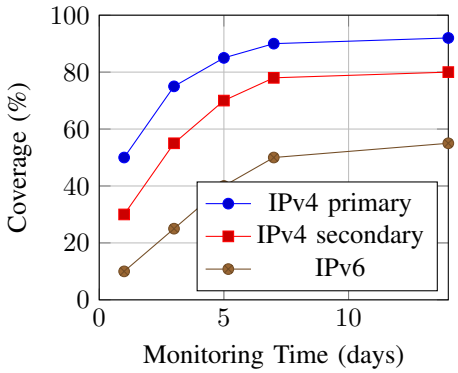

Fig. 2. Host Coverage over Time.

The twin built by NotLine achieved approximately an 85% node coverage within the primary subnet according to the set of hostnames and IP addresses in the flow data. Critical infrastructure nodes show a consistent presence in the dataset. Their regular communications via broadcast and multicast protocols ensure their representation in the digital twin.

Figure 2 shows the percentage of unique hosts NotLine discovered over a 14-day monitoring period, separately for the primary IPv4 subnet (131.114.x.x), the secondary IPv4 subnet (146.48.x.x), and the IPv6 segment.

In the first three days, ARP and DHCP-based discovery rapidly identify more than 70% of devices in the primary IPv4 range, whereas the secondary IPv4 and IPv6 segments lag due to lower initial lease churn and multicast activity, respectively.

Between days 4 and 7, service-discovery protocols such as mDNS and SSDP contribute additional host revelations, boosting coverage from approximately 85% to 90% in the primary IPv4 subnet, from 70% to 78% in the secondary IPv4, and from 40% to 50% in IPv6.

After day 7, the curves plateau: new hosts appear only sporadically, yielding marginal gains below 5% despite continued monitoring. This indicates that a 5–7 day monitoring window achieves a near-complete inventory of network nodes with efficient use of sensor resources.

Node visibility varies significantly by protocol, with service discovery protocols (MDNS, SSDP) providing the broadest node coverage (capturing 92% of networked devices), while application-specific protocols (SSH, Dropbox) reveal specialized subsets of the network population.

D. Key Findings

Experimental results show that to build an accurate and reliable twin of the experimented network NotLine requires:

- At least about 250 messages per monitored network, with representation across at least 4 distinct protocol types.
- A persistence time of at least 5-7 days to capture both daily and weekly network behaviour patterns.
- Monitoring strategies that prioritize both high-volume server connections and broadcast/multicast protocols to achieve comprehensive node coverage.

These findings highlight the importance of balancing the duration of monitoring, the diversity of protocols and node coverage when designing a security twin. The analysis confirms that digital twins built by monitoring flow data can lead to an accurate twin with a large coverage, provided sufficient message volume and persistence time.

VII. A Steep-then-Slow Discovery Pattern

Let us consider now a model of the discovery of new nodes. Experimental observations reveal that most nodes are discovered in two days. Thereafter, there is a significantly slower decay in discovery rate. An accurate model of this characteristic "steep drop followed by long tail" behaviour is a hypoexponential function [22]:

$$f(t) = \alpha_1 e^{-\beta_1 t} + \alpha_2 e^{-\beta_2 t},$$

Where:

- α_1 and β_1 determine the initial amplitude of the fast-decay component ($\beta_1 \gg \beta_2$),
- α_2 and β_2 govern the slower long-tail decay behavior.

After fitting the data collected up to the present, we have determined the following parameter estimates:

$$\hat{\alpha}_1 = 100, \quad \hat{\beta}_1 = 3, \quad \hat{\alpha}_2 = 25, \quad \hat{\beta}_2 = 0.2,$$

resulting in an initial discovery rate of $f(0) = \hat{\alpha}_1 + \hat{\alpha}_2 = 125$ new items per day. The model provides a good fit to the observed data, with a determination coefficient $R^2 = 0.76$, indicating that the hypoexponential model explains 76% of the variance in discovery rates.

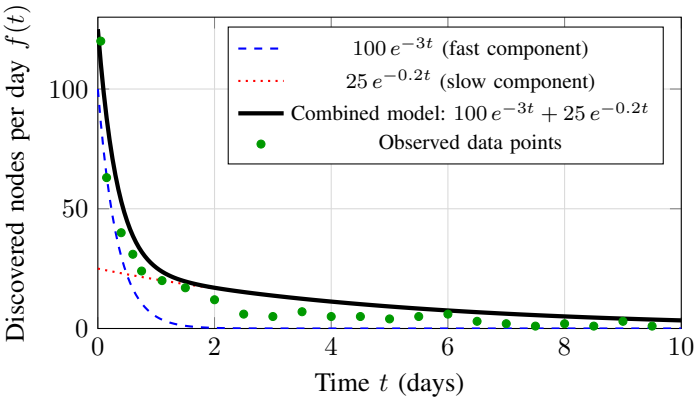

Fig. 3. Hypoexponential model with observed data: The steep initial decline (blue dashed line) combined with a slower long-tail decay (red dotted line) produces the overall discovery pattern (solid black line). Green points show actual measured discovery rates that validate the model fit ($R^2 = 0.76$).

Figure 3 shows that by approximately two days, the fast component $100e^{-3t}$ has already decreased by more than 90%, while the slow component $25e^{-0.2t}$ becomes predominant thereafter. This model effectively captures both the dramatic early decline and the gradual tapering observed over extended periods. This hypoexponential pattern resembles worm propagation curves in hierarchical network systems [21], where worm spreading similarly exhibits an initial steep growth

phase followed by a fairly slower long-tail progression. This characteristic pattern emerges because reaching some nodes requires a long time due to the low number of network interactions of these nodes.

VIII. Conclusions

NotLine is a fully automated *non-intrusive* platform for building and updating a security twin through continuous passive network monitoring. By relying solely on metadata captured by ntopng across multiple protocols, NotLine avoids the risks and overhead of active scanning while delivering a dynamically synchronized representation of an ICT infrastructure.

Our research shows how passive feeds produced by the passive monitoring of networks protocols can be merged into a single metadata stream that NotLine maps to an evolving network model through filtering, normalization, and graph-based correlation, automatically identifying and mapping both devices and services. Our experiments on a medium-sized LAN at a University department revealed that the building of an accurate twin requires the capture of at least *250 messages* spanning four protocol families and a continuous monitoring window of *5–7 days* to capture usage patterns and ensure high network coverage. NotLine's fully automated pipeline minimizes human intervention and quickly handles topology changes. Overall, NotLine advances proactive cybersecurity by supporting a live, detailed, and actionable model of network infrastructure, without the drawbacks of active scanning.

An important current limitation of NotLine is that its passive approach cannot guarantee complete visibility of all network assets, particularly those with minimal communication patterns or employing proprietary protocols. Furthermore, while effectively mapping the topology, NotLine provides limited insight into the frequency of communication.

Future work will explore the integration of host-level telemetry and endpoint sensors to enrich the twin with process- and application-level insights, real-time ingestion of external threat-intelligence feeds for adaptive monitoring, and reinforcement-learning agents that autonomously recommend and validate remediations.

References

[1] Blake Anderson and David McGrew. "OS Fingerprinting: New Techniques and a Study of Information Gain and Obfuscation". In: *arXiv preprint arXiv:1706.08003*. 2017.

[2] F. Baiardi, S. Ruggieri, and V. Sammartino. "AI-enabled Cybersecurity using Synthetic Data". In: *Digital Twins Ecosystems and Applications, PerCom 2025*. 2025.

[3] F. Baiardi, S. Ruggieri, and V. Sammartino. "Anticipating Disasters through a Security Twin". In: *ARES - DOD 2024*. 2024.

[4] Fabrizio Baiardi and Vincenzo Sammartino. "A Quantitative Framework for the Validation of Twin-Based Cyber Defense". In: *I3M*. 2025.

[5] Gene Bartlett and John Heidemann. "Understanding Passive and Active Service Discovery". In: *Proceedings of the 7th ACM SIGCOMM*. 2007, pp. 1–14.

[6] CyCognito. *Active vs Passive Reconnaissance: 6 Key Differences*. 2023.

[7] Rasmus Dahlberg et al. "Digital twin as risk management: a process perspective". In: *International Journal of Disaster Risk Reduction* 39 (2019), p. 101262.

[8] Luca Deri, Maurizio Martinelli, and Alfredo Cardigliano. "nDPI: Open-source high-speed deep packet inspection". In: *2014 IWCMC*. IEEE. 2014, pp. 1–6.

[9] Amy Fuller et al. "Digital twin: Enabling technologies, challenges and open research". In: *IEEE Access* 8 (2020), pp. 108952–108971.

[10] Steven Furnell et al. "Understanding the full cost of cyber security breaches". In: *Computer fraud & security* 12 (2020), pp. 6–12.

[11] ICS Cybersecurity Conf. *Active vs. Passive Network Monitoring: No Longer an Either-Or Proposition*. 2019.

[12] Rahim Khan et al. "Artificial intelligence-enabled digital twin framework for cybersecurity risk assessment". In: *2022 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE. 2022, pp. 1–8.

[13] Quanyan Li et al. "Monte Carlo-based threat assessment of power system security". In: *2020 PMAPS*. IEEE. 2020, pp. 1–6.

[14] NIST. *National Vulnerability Database*. 2024.

[15] runZero. *Active Scanning Industrial Control Systems Safely*. 2024.

[16] A. Saleh and R. Ahmed. "Two decades of cyberattack simulations: A systematic literature review". In: *Journal of Network and Computer Applications* 200 (2022), p. 103345.

[17] V. Sammartino. "A Framework for Proactive Cyber-Resilience: Non-Intrusive Modeling for Autonomous Defense". In: *DS-RT 2025*. 2025.

[18] V. Sammartino, F. Baiardi, and S. Ruggieri. "A Security Twin to Defeat Intrusions in Cyber Physical Systems". In: *ESREL SRA-E 2025*. 2025.

[19] Omer San, Adil Rasheed, and Trond Kvamsdal. "Hybrid analysis and modeling, eclecticism, and multifidelity computing toward digital twin revolution". In: *GAMM-Mitteilungen* 44.2 (2021), e202100007.

[20] The MITRE Corporation. *CVE*. 2024.

[21] Tianbo Wang and Chunhe Xia. "H2P: A Novel Model to Study the Propagation of Modern Hybrid Worm in Hierarchical Networks". In: *Algorithms and Architectures for Parallel Processing*. 2020, pp. 251–269.

[22] George P. Yanev. "Exponential and Hypoexponential Distributions: Some Characterizations". In: *Mathematics* 8.12 (2020), p. 2207.

[23] Zirui Zhao, W. Lee, and David Hsu. "LLMs as Commonsense Knowledge for Large-Scale Task Planning". In: *ArXiv* abs/2305.14078 (2023).