

Article

Auditing Inferential Blind Spots: A Framework for Evaluating Forensic Coverage in Network Telemetry Architectures

Mehrnoush Vaseghipanah ¹, Sam Jabbehdari ^{1,*} and Hamidreza Navidi ²

¹ Department of Computer Engineering, NT.C., Islamic Azad University, Tehran 1651153511, Iran; mehrnoush.vaseghipanah@iau.ac.ir

² Department of Mathematics and Computer Sciences, Shahed University, Tehran 3319118651, Iran; navidi@shahed.ac.ir

* Correspondence: sam.jabbehdari@iau.ac.ir

Abstract

Network operators increasingly rely on abstracted telemetry (e.g., flow records and time-aggregated statistics) to achieve scalable monitoring of high-speed networks, but this abstraction fundamentally constrains the forensic and security inferences that can be supported from network data. We present a design-time audit framework that evaluates which threat hypotheses become non-supportable as network evidence is transformed from packet-level traces to flow records and time-aggregated statistics. Our methodology examines three evidence layers (L0: packet headers, L1: IP Flow Information Export (IPFIX) flow records, L2: time-aggregated flows), computes a catalog of 13 network-forensic artifacts (e.g., destination fan-out, inter-arrival time burstiness, SYN-dominant connection patterns) at each layer, and maps artifact availability to tactic support using literature-grounded associations with MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK). Applied to backbone traffic from the MAWI Day-In-The-Life (DITL) archive, the audit reveals selective inference loss: Execution becomes non-supportable at L1 (due to loss of packet-level timing artifacts), while Lateral Movement and Persistence become non-supportable at L2 (due to loss of entity-linked structural artifacts). Inference coverage decreases from 9 to 7 out of 9 evaluated ATT&CK tactics, while coverage of defensive countermeasures (MITRE D3FEND) increases at L1 (7 → 8 technique categories) then decreases at L2 (8 → 7), reflecting a shift from behavioral monitoring to flow-based controls. The framework provides network architects with a practical tool for configuring telemetry systems (e.g., IPFIX exporters, P4 pipelines) to reason about and provision the minimum forensic coverage.

Keywords: network telemetry; evidence abstraction; forensic inference; MITRE ATT&CK; MITRE D3FEND; threat intelligence



Academic Editors: Hakim Mellah and Filippo Malandra

Received: 24 December 2025

Revised: 24 January 2026

Accepted: 27 January 2026

Published: 29 January 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Network telemetry has evolved from simple counters to rich flow records and programmable data-plane exports, enabling scalable monitoring of high-speed networks [1]. This abstraction is driven by practical constraints: storage costs, processing overhead, and privacy regulations make continuous packet-level monitoring infeasible for most organizations [2]. Consequently, modern network security operations rely heavily on aggregated telemetry—flow records, time-binned statistics, and log summaries—rather than full packet capture.

However, this evolution introduces a fundamental trade-off: telemetry abstraction, while necessary for scalability, silently removes forensic evidence required for threat inference [3]. The transition from packet-level to flow-level and time-aggregated representations fundamentally alters the observable evidence available for forensic analysis and threat reasoning. This challenge is compounded by the widespread adoption of encryption: Transport Layer Security (TLS) 1.3 encrypts larger fractions of the handshake, further reducing semantic visibility and shifting feasible reasoning toward metadata and behavioral artifacts [4]. The ENISA Threat Landscape 2025 reports that a substantial fraction of observed incidents are categorized as “unknown,” reflecting limitations in incident reporting, sector attribution, and outcome visibility in open-source and shared data [5]. These documented gaps motivate a closer examination of how evidence abstraction and monitoring practices constrain defensible forensic inference in network telemetry architectures. While prior work has extensively studied detection accuracy across different data representations [6,7], a critical gap remains: how does evidence abstraction constrain the defensibility of forensic threat inference, independent of detection or classification accuracy?

To illustrate this conceptual distinction, consider a concrete scenario: an incident responder analyzing archived network telemetry to investigate a potential lateral movement incident. A detection system may correctly identify suspicious traffic patterns with high accuracy, but if the archived telemetry consists only of time-aggregated flow statistics (L2), the responder cannot defensibly support a Lateral Movement hypothesis because the required artifact (destination fan-out, which requires per-entity source-destination pairs) is not computable from aggregated statistics. The detection system’s accuracy is irrelevant if the evidence representation does not support the forensic inference required for incident response. This example demonstrates that detection accuracy (measuring how well a system identifies threats) and defensibility of inference (measuring whether the evidence representation supports forensic claims) are independent concerns: high detection accuracy does not guarantee defensible inference capability.

This paper introduces a design-time audit framework that, given a telemetry schema (e.g., NetFlow, time-aggregated flows), identifies which MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) tactic-level hypotheses are no longer supported due to lost evidence. Our backbone network monitoring context assumes a passive observer with access to packet headers captured at a transit network, without payload visibility, endpoint context, or ground-truth labels. This setting reflects ISP/carrier monitoring realities and constrains the analysis to network-level behavioral patterns observable without endpoint logs or payload data. We restrict analysis to network-observable tactics (e.g., Command and Control, Exfiltration), excluding endpoint-dependent tactics (e.g., Credential Access) per our backbone threat model. This ensures our audit reflects the realistic capabilities of network-layer telemetry.

For incident responders, this gap manifests as a critical dilemma: Analysts must reason about adversary behavior from partial, archived network evidence, but the challenge is not merely detecting known attack patterns. Instead, they must determine which threat hypotheses can be logically supported given the available evidence representation. For example, does the available evidence representation permit supporting a claim that observed traffic patterns indicate Lateral Movement when only time-aggregated flow statistics are available? Unlike detection-focused studies that measure false-positive rates, our framework evaluates whether a given telemetry layer logically supports or refutes a threat hypothesis (e.g., “This traffic pattern is consistent with Lateral Movement”) based on available evidence artifacts—regardless of the detection algorithm used.

This gap has practical consequences. Security teams may invest in monitoring infrastructure without understanding which classes of forensic inference become non-supportable

under their chosen abstraction level. Incident response playbooks may include procedures that require evidence types no longer available in archived telemetry. Without a systematic audit, organizations cannot align their defensive strategies with their actual forensic capabilities.

The importance of this challenge extends beyond individual organizations to the broader network security ecosystem. Security Operations Centers (SOCs) and incident response teams routinely face scenarios where archived telemetry must support forensic reconstruction of adversary activities, but the available evidence representation may not support the required inferences. This problem is particularly acute in backbone and ISP monitoring contexts, where scalability constraints force operators to rely on abstracted telemetry, yet security requirements demand forensic capabilities for threat investigation and incident response. The relationship between telemetry design choices and forensic capabilities is not well understood: operators may configure monitoring systems for efficiency without realizing which threat hypotheses become non-supportable under their chosen abstraction level.

The practical challenges are multifaceted. Network operators must balance competing requirements: scalability demands abstraction (flow records, time-aggregated statistics), while security operations require forensic capabilities (packet timing, entity structure). This tension is compounded by the increasing adoption of encryption, which further reduces semantic visibility and shifts feasible reasoning toward metadata and behavioral artifacts. Additionally, the widespread use of programmable data planes (e.g., P4) and software-defined networking (SDN) introduces new opportunities for telemetry configuration, but also new complexity in understanding the forensic implications of design choices. Without systematic methods to evaluate these trade-offs, operators may make suboptimal decisions that compromise forensic capabilities without fully understanding the consequences.

Research Overview and Contributions

We present a methodological framework for design-time audit that evaluates which forensic claims remain supportable under a given telemetry architecture. The contribution is methodological: the audit procedure evaluates representational limits of inference rather than proposing a new detection or classification algorithm. We develop a reproducible audit methodology that examines the limits of inference that arise when network evidence is reduced through aggregation and abstraction. Our framework uses MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) and Detection, Denial, and Disruption Framework Emulating Network Defense (D3FEND)—the de facto industry standards for structured threat and defense modeling—as vocabularies for organizing defensive reasoning, treating them as hypothesis spaces rather than classification taxonomies. While ATT&CK tactics map to adversary behaviors observable at specific evidence layers (e.g., L0 packet timing for “Execution”), D3FEND countermeasures map to defensive actions enabled by those layers (e.g., L1 flow records enable “Network Traffic Throttling” against DDoS). Our audit characterizes how abstraction shifts the actionable defense portfolio. This makes the framework’s outputs directly relevant for security engineers designing telemetry systems. From a networking perspective, the proposed audit framework is intended to support the design and evaluation of network monitoring architectures, rather than post-hoc forensic reconstruction. It provides a systematic method for reasoning about which network-level inference classes remain feasible under specific telemetry export and aggregation choices.

Using backbone traffic from the MAWI DITL archive [8] and a literature-grounded inference framework, our audit identifies inference loss that is (a) selective (affecting specific tactics) and (b) structural (erasing relational evidence like host fan-out). Our

audit identifies evidence abstraction as a source of unfalsifiable hypotheses—where critical artifacts (e.g., packet timing) are removed, rendering certain threat claims neither provable nor disprovable. The audit reveals that (i) packet-level timing artifacts required to support Execution hypotheses disappear once monitoring collapses to flows, (ii) entity-linked structural artifacts such as destination fan-out and host-specific periodicity are lost under time-aggregated telemetry, eliminating supportable claims for Lateral Movement and Persistence, and (iii) defensive reasoning does not degrade monotonically with abstraction, but instead transforms the supportable defensive toolkit from behavioral monitoring to rate- and flow-based controls.

Research Contributions: Our contributions include the following:

1. Design-time audit framework: We provide a design-time audit tool for network architects to evaluate and configure telemetry systems (e.g., IPFIX exporters, P4 pipelines) to reason about and provision minimum forensic coverage. We define evidence loss as a reduction in actionable forensic options: when telemetry abstraction removes critical artifacts, certain threat hypotheses become unfalsifiable—neither provable nor disprovable from the available data.
2. Structured threat and defense modeling: We restrict analysis to tactic-level inference using ATT&CK as a hypothesis space and D3FEND as defensive applicability, providing a structured vocabulary for reasoning about defensive capabilities under partial observability. This approach enables systematic evaluation of forensic coverage using industry-standard frameworks.
3. Selective and structural inference loss: We demonstrate that inference collapse is selective and structural rather than uniform, with defensive reasoning transforming rather than uniformly degrading across abstraction levels. This finding informs the design of hybrid monitoring systems that strategically combine different abstraction levels to maximize forensic coverage.
4. Backbone/ISP monitoring context: We position the analysis in backbone/ISP monitoring contexts, making the work directly applicable to network-wide security service management and programmable data plane telemetry design. The framework addresses the specific constraints and requirements of large-scale network monitoring.

We review related work on network forensics and evidence abstraction (Section 2), present our audit methodology (Section 3), report results from backbone traffic analysis (Section 4), discuss implications for network monitoring architecture design (Section 6), and synthesize implications for practice (Section 5).

2. Related Work

2.1. Backbone and ISP-Level Network Visibility

A key determinant of what can be inferred from network observations is the vantage point and the data product available to the analyst. Longitudinal and large-scale Internet Service Provider (ISP) studies illustrate that backbone monitoring is typically conducted using flow records rather than full packet capture, due to throughput, storage, and privacy constraints. Trevisan et al. analyze the multi-year evolution of Internet usage from a national ISP using rich flow-level measurements, showing both the feasibility and the inherent abstraction of ISP-scale observation (e.g., service trends and protocol evolution rather than payload semantics) [1]. Benes et al. similarly emphasize that high-speed ISP backbones are commonly monitored via IP flows; even then, long-term datasets are difficult to retain and “brief” summary statistics can be insufficient for deeper analyses without further processing [9].

Other backbone/Internet Exchange Point (IXP) studies demonstrate that even when packet-level traces are available, interpretability remains bounded by what can be reliably observed without endpoint context. Maghsoudlou et al. dissect Internet traffic using port 0 across complementary packet- and flow-level datasets and show that substantial apparent “anomalies” can stem from artifacts such as fragmentation, while only limited subsets provide payload-bearing evidence [10]. Collectively, these works motivate a backbone-forensics framing in which the analyst must reason from incomplete, sometimes artifact-prone observations, and where evidence is frequently available only in aggregated forms.

A complementary line of work highlights that, at ISP scale, even distinguishing benign background noise from operationally meaningful signals can require carefully designed tests under constrained observability. Gigis et al. propose Penny, an ISP-deployable test to differentiate spoofed aggregates from non-spoofed traffic that enters at unexpected locations by exploiting retransmission behavior after selectively dropping a small number of TCP packets [11]. While motivated by operational alerting, the key methodological relevance for backbone forensics is that inference may depend on what is observable and repeatable under transit constraints, and that seemingly anomalous aggregates can remain ambiguous without additional evidence channels or carefully bounded interventions.

Backbone datasets targeting encrypted traffic further illustrate how ISP visibility is often mediated through derived products rather than raw payloads. Hynek et al. introduce CESNET-TLS-Year22, a year-spanning dataset captured on 100 Gbps backbone links, provided as Transport Layer Security (TLS)-relevant flow representations augmented with limited early-connection packet sequences, packet histograms, and selected fields from the TLS ClientHello [12]. Although created primarily to support traffic classification research, the dataset design is directly relevant to our framing: it exemplifies the practical evidence surface available at scale for encrypted traffic—metadata and partial handshake-derived artifacts—while underscoring the absence of payload semantics and the resulting need for conservative, uncertainty-aware forensic reasoning.

2.2. Evidence Transformation Through Flow Construction

Transforming packets into flows is not a neutral preprocessing step; it is an evidence transformation that can change what conclusions remain defensible. Flow monitoring configuration choices—especially expiration timeouts—can split, merge, or distort activity patterns that might otherwise be interpretable. Velan and Jirsik explicitly demonstrate that the configuration of flow monitoring affects the resulting flow records and can materially impact downstream analytics, using Slowloris as an example of the sensitivity to timeout choices [3].

At the ISP scale, additional uncertainty arises because flow-derived inferences may rely on indirect reasoning from partial telemetry. Schou et al. consider ISP flow analytics under measurement noise and infer flow splitting ratios indirectly from observed demands and link utilization, reflecting the practical reality that operators often cannot directly observe the internal determinants of traffic distribution [13]. In a complementary operational direction, Flowyager addresses the distributed nature and volume of flow records by constructing compact summaries (Flowtrees) to support interactive, network-wide queries [14]. While these systems are not designed for forensic inference per se, they underline a central point for evidence-centric reasoning: the construction, configuration, and summarization of flow data define the boundaries of what can later be reconstructed regarding temporal ordering, causality, and behavioral structure.

2.3. Time Aggregation, Sampling, and Telemetry Abstraction

Beyond flows, monitoring pipelines commonly introduce additional abstraction through time binning, sampling, and sketching, each of which trades semantic detail for scalability. Magnifier illustrates the operational dependence on sampling for global ISP monitoring and proposes complementing sampling with mirroring to improve coverage without prohibitive overhead [15]. Du et al. study sampling at the per-flow level and propose self-adaptive sampling to allocate measurement effort unevenly across flows, reflecting that practical measurement policies are rarely uniform and can differentially preserve evidence across traffic classes [16]. Operational NetFlow pipelines introduce additional temporal abstraction through exporter-driven record generation and reporting delays, which can materially affect what can be inferred when from flow evidence. He et al. analyze these delays in the context of NetFlow-based ISP Distributed Denial of Service (DDoS) monitoring and propose FlowSentry, which leverages sketch-based sliding windows and cross-router correlation to reason over partially reported flow records [17]. While the objective in that work is accelerated detection, its methodological significance for our study is different: it makes explicit that time-windowing and incremental reporting are intrinsic properties of flow telemetry at scale, and thus that downstream reasoning must treat flow and time-aggregated flow products as evidence transformed by the monitoring pipeline, not as faithful surrogates for packet-level behavior.

A broad telemetry literature further formalizes these abstractions through compact data structures and programmable data planes. Landau-Feibish et al. survey compact data structures and streaming methods for telemetry in programmable devices, emphasizing the tight memory/compute constraints that force approximate summaries and selective retention [18]. Several works address the mechanics of time-windowed telemetry—an issue directly relevant to time-aggregated flow evidence. OmniWindow proposes fine-grained sub-windows that can be merged into multiple window types under switch resource constraints [19]. Namkung et al. show that practical telemetry can suffer from accuracy degradation due to delays when pulling and resetting state, emphasizing that implementation and retrieval workflows shape the fidelity of exported summaries [20]. SketchPlan and AutoSketch raise the abstraction layer for deploying sketch-based telemetry and compiling high-level intents into sketches, again foregrounding that operators ultimately receive derived artifacts rather than raw observations [21,22]. SetD4 extends data-plane set representations with deletion and decay, explicitly embedding time-based forgetting into telemetry structures [23]. F3 explores split execution between ASICs and FPGAs to enable richer monitoring patterns under throughput constraints [24]. Liu et al. discuss ecosystem-level challenges that hinder the adoption of sketch-based telemetry despite its theoretical appeal, reflecting a persistent gap between what is desirable for visibility and what is feasible at scale [25].

Long-horizon ISP datasets further illustrate how operational constraints steer evidence products toward time-aggregated abstractions rather than packet- or even flow-complete archives. CESNET-TimeSeries24 provides 40 weeks of time-series traffic statistics derived from an ISP network at the scale of hundreds of thousands of active IP addresses, explicitly motivated by the lack of long-term real-world datasets for forecasting and anomaly analysis and the risk of overestimating conclusions when evaluations rely on synthetic or short-window traces. While the dataset is introduced to support modeling and anomaly/forecasting research, its design choices are directly relevant to backbone forensics: the exported representation is already a temporally aggregated statistical view, which inherently limits the reconstructability of fine-grained behavioral evidence and makes inference dependent on which summary metrics survive the aggregation pipeline [26]. These contributions collectively motivate treating packet → flow → time-aggregated (and

sketch/sampling-based) transformations as a progressive reduction in evidential granularity. They provide concrete technical mechanisms—windowing, decay, sketch compilation, sampling—that explain how inferential capacity can collapse even when monitoring remains operationally effective.

2.4. Architectures Balancing Packet Fidelity and Scalability

Several monitoring architectures explicitly try to balance packet-level fidelity with scalable operation, acknowledging that the choice is not binary. FloWatcher-DPDK demonstrates high-speed software monitoring that can provide tunable statistics at the packet and flow levels, illustrating a practical continuum between detailed and summarized evidence products [27]. MONA introduces adaptive measurement that reduces task sets under bottlenecks to maintain monitoring objectives, operationalizing a dynamic reduction of observed detail when conditions demand it [28]. FlexMon and FPGA-based flow monitoring systems aim to provide fine-grained measurement under strict resource constraints in programmable/hardware settings [29,30]. Recent work on programmable data planes further underscores that the monitoring substrate can shape the evidentiary record by deciding which packet-level properties to preserve and which to compress into statistics. Doriguzzi-Corin et al. propose P4DDLe, using P4-programmable switches to selectively extract raw packet features (including categorical features) and to organize them so that aspects of flow semantics are preserved under resource constraints [31]. Although positioned toward Network Intrusion Detection System (NIDS) pipelines, the relevant implication for our paper is architectural: it illustrates a principled attempt to mitigate semantic loss introduced by flow/statistical compression, reinforcing our premise that packet→flow→aggregated-flow transformations should be treated as progressive reductions in evidential granularity with direct consequences for what inferences remain defensible.

HybridMon directly targets the tension between flow efficiency and packet-level usefulness by combining condensed packet-level monitoring with selective flow aggregation in programmable switches [32]. Hardegen’s scope-based monitoring highlights that even within “flow monitoring,” the analyst may enlarge the scope (e.g., bidirectional context, subflows in time windows) to regain granularity, at the cost of overhead [33]. Although these works are often motivated by operational monitoring or intrusion detection pipelines, their relevance here is methodological: they show that modern infrastructures deliberately shape the evidentiary record by selectively exporting different representations over time and under load—exactly the conditions under which forensic reasoning must quantify what inferences remain defensible.

Comparison with Our Work: The architectures discussed above (FloWatcher-DPDK, MONA, FlexMon, P4DDLe, HybridMon, scope-based monitoring) focus on how to design telemetry systems that balance packet-level fidelity with scalable operation. They provide design mechanisms such as tunable statistics, adaptive measurement, selective packet extraction, and hybrid configurations. In contrast, our audit framework focuses on evaluating which forensic inferences remain supportable under given telemetry configurations, regardless of how those configurations were designed. While prior works optimize for operational efficiency and detection performance, our framework evaluates the representational limits of inference, independently of detection algorithms. Prior works answer “how should we design telemetry systems?” while our framework answers “which threat hypotheses can be supported given a telemetry configuration?” These approaches are complementary: prior works provide design mechanisms for creating telemetry systems, while our framework provides evaluation criteria for assessing their forensic coverage. Network architects can draw on prior work to design telemetry systems, then use our framework to evaluate which forensic capabilities those systems preserve.

2.5. Encrypted Traffic and Forensic Constraints

The growth of encryption further reduces semantic visibility and shifts feasible reasoning toward metadata, timing, and behavioral artifacts. Surveys of encrypted traffic analysis characterize the space of approaches and their dependence on what parts of the protocol remain observable. Papadogiannaki and Ioannidis survey applications, techniques, and countermeasures in encrypted traffic analysis, emphasizing both the feasibility of inference from encrypted traces and the privacy-driven limitations and evasion dynamics that constrain what can be concluded [34]. Sharma and Lashkari provide a more recent survey focused on identification/classification techniques and challenges in encrypted traffic, reflecting the broader research emphasis on learning-based inference from metadata while also noting dataset and operational constraints [35].

TLS 1.3 intensifies this trend by encrypting a larger fraction of the handshake, motivating new methods and highlighting the limitations of older approaches. Zhou et al. survey TLS 1.3 encrypted traffic analysis, reviewing the impact of TLS 1.3 features (e.g., 0-RTT, PFS, ECH) and cataloging families of analysis methods and datasets [4]. At a more application-specific forensic level, Sarhan et al. propose a framework for digital forensics of encrypted real-time traffic in messaging and Voice over IP (VoIP) contexts, aiming to extract user behavior from encrypted traces [36]. Notably, such work often assumes the availability of traces and features, including application-specific patterns and, in some settings, deeper inspection capabilities than backbone monitoring typically affords.

Within backbone-forensics constraints—packet headers without payload and no endpoint ground truth—these surveys and frameworks are most relevant for establishing why the inference problem is structurally underdetermined and why conservative, uncertainty-aware reasoning is necessary when evidence is reduced from packets to flows and then time-aggregated flows.

2.6. ATT&CK and D3FEND as Reasoning Frameworks

ATT&CK and D3FEND are frequently used to organize threat knowledge and defensive measures, but their role varies substantially across studies. Al-Sada et al. provide a comprehensive survey of how ATT&CK has been leveraged across sectors and methodologies, offering a useful reference for the breadth of ATT&CK-based work and the variety of assumptions and auxiliary data employed [37]. This diversity is central to evidence-centric analysis: many ATT&CK use cases implicitly rely on rich visibility (endpoint telemetry, payload inspection, curated labels, or incident reports) that may not hold in backbone settings.

Several works combine ATT&CK with D3FEND to connect representations of adversary behavior with defensive actions. Yousaf and Zhou demonstrate ATT&CK/D3FEND modeling in a maritime cyber-physical scenario and propose defensive mechanisms informed by the frameworks [38]. Vaseghipanah et al. integrate ATT&CK and D3FEND into a game-theoretic model for digital forensic readiness, mapping Advanced Persistent Threat (APT) behaviors to defensive countermeasures and deriving strategic recommendations under uncertainty [39]. These studies highlight the utility of ATT&CK/D3FEND as organizing structures for reasoning about threats and defenses, but they often operate at a level where attacker–defender modeling, domain context, or expert weighting supplies additional semantics beyond what passive backbone artifacts can justify.

A further methodological consideration is the common tendency in parts of the broader literature to treat ATT&CK at the technique level as a labeling scheme (e.g., as targets for detection or classification). In contrast, the present paper’s alignment is closer to survey- and modeling-oriented uses that treat ATT&CK primarily as a vocabulary for structured reasoning. By restricting ATT&CK usage to the tactic level and grounding any artifact–tactic support exclusively in published descriptions, the analysis aims to remain

faithful to what can be justified under partial observability. Similarly, D3FEND is used here to reason about defensive applicability—which defensive technique categories remain logically actionable given the evidence product—rather than to evaluate mitigation success or deployment efficacy, thereby maintaining separation between evidential support and operational effectiveness.

Comparison with Our Work: The ATT&CK/D3FEND-based works discussed above ([38,39]) use these frameworks for strategic threat and defense modeling with rich visibility assumptions. They operate at the technique level with additional semantics such as game-theoretic modeling, expert weighting, domain context (maritime cyber-physical systems), and attacker–defender dynamics. These approaches evaluate defensive effectiveness and derive strategic recommendations for deployment. In contrast, our work uses ATT&CK/D3FEND for evidence-centric reasoning under partial observability constraints (backbone monitoring, no endpoint context, no payload visibility). We restrict analysis to the tactic level and ground artifact–tactic associations exclusively in published literature, remaining faithful to what can be justified from network-level observations alone. While prior works assume rich visibility that enables technique-level attribution and strategic modeling, our work evaluates which tactics remain supportable given available evidence representations. Prior works answer “what defensive strategies should be deployed?” while our work answers “which threat hypotheses can be supported given available evidence?” This distinction is crucial: prior works optimize defensive deployment under rich visibility, while our work evaluates inference supportability under partial observability, making it directly applicable to backbone/ISP monitoring contexts where endpoint logs and payload access are unavailable.

2.7. Gap and Contribution

The reviewed literature demonstrates that (i) backbone monitoring relies on abstracted telemetry (flow records, time-aggregated statistics), (ii) evidence transformation through flow construction and temporal aggregation affects what can be inferred, and (iii) ATT&CK and D3FEND provide structured vocabularies for organizing threat reasoning and defensive actions. However, prior work has not explicitly provided a reproducible, representation-driven audit that determines which ATT&CK tactic-level hypotheses remain defensibly supportable under progressively abstracted network telemetry, nor a framework for evaluating telemetry architectures based on forensic coverage.

Prior work focuses on (i) detection accuracy under different data representations (exemplified by empirical studies comparing packet- vs. flow-based detection [6,7]), (ii) operational efficiency of telemetry systems [14,15,32], (iii) dataset creation for traffic analysis [12,26], or (iv) strategic modeling using ATT&CK/D3FEND with rich visibility (endpoint logs, payload access, or domain context) [38,39]. Comprehensive surveys of encrypted traffic analysis [4,34,35] catalog techniques and challenges but do not address how evidence abstraction constrains forensic defensibility. Surveys of DDoS defense mechanisms [2] and ATT&CK usage across sectors [37] similarly focus on detection effectiveness or strategic deployment rather than inference supportability under partial observability. Case studies of digital forensics frameworks [36] demonstrate forensic reconstruction capabilities but assume richer evidence availability (payload access, endpoint context) than backbone monitoring typically affords. However, we did not find prior work that addresses the defensibility of forensic inference under partial observability or provides a systematic audit of which threat hypotheses become non-supportable as evidence is abstracted.

What we add: We provide a design-time audit framework that (i) systematically evaluates which MITRE ATT&CK tactic-level hypotheses become non-supportable as evidence is transformed from packets (L0) to flows (L1) to time-aggregated statistics (L2), (ii) maps

artifact availability to tactic support using literature-grounded associations, (iii) quantifies defensive applicability using D3FEND under backbone-forensics constraints (no payload, no endpoint context, no ground truth), and (iv) demonstrates that inference collapse is selective and structural rather than uniform, with defensive reasoning transforming rather than uniformly degrading. In practice, the audit produces a "forensic coverage report" that can guide IPFIX schema choices, exporter configurations (timeouts/binning), and programmable telemetry designs. This framework enables network architects to evaluate telemetry designs and configure systems (e.g., IPFIX exporters, P4 pipelines) to reason about and provision minimum forensic coverage. Table 1 summarizes representative related work across evidence products, vantage points, and primary goals.

Table 1. Comparison of related work on network forensics and telemetry abstraction. Evidence products: PCAP headers (packet-level), Flow (bidirectional flow records), Time-series (aggregated statistics), Hybrid (selective packet/flow), Sketch (approximate summaries). Primary goals: Characterization (traffic analysis), Telemetry system (monitoring architecture), Detection (threat detection/monitoring), Operational Test (ISP-deployable testing), Dataset (data collection/curation), Survey (literature review), Modeling (threat/defense modeling), Forensic audit (supportability evaluation).

Work	Evidence Product	Vantage	Primary Goal	ATT&CK/D3FEND	Addresses Supportability?
Trevisan et al. (2018) [1]	Flow	Backbone/ISP	Characterization	None	×
Velan and Jirsik (2020) [3]	Flow	Network monitoring	Telemetry system	None	×
Flowyager (Saidi et al., 2020) [14]	Flow	Network monitoring	Telemetry system	None	×
Gigis et al. (2024) [11]	Flow / Active Test	Backbone/ISP	Operational Test	None	×
FlowSentry (He et al. (2025)) [17]	Flow	Backbone/ISP	Detection	None	×
P4DDLe (2024) [31]	Hybrid	Programmable switches	Telemetry system	None	×
HybridMon (Fink et al., 2025) [32]	Hybrid	Programmable switches	Telemetry system	None	×
Papadogiannaki and Ioannidis (2021) [34]	PCAP/Flow	Various	Survey	None	×
CESNET-TimeSeries24 (Koumar et al., 2025) [26]	Time-series	Backbone/ISP	Dataset	None	×
Yousaf and Zhou (2024) [38]	Various	Enterprise/CPS	Modeling	Both (technique)	×
Vaseghipanah et al. (2025) [39]	Various	Enterprise	Modeling	Both (technique)	×
This work	L0/L1/L2	Backbone/ISP	Forensic audit	Both (tactic)	✓

"Supportability" refers to whether a given telemetry representation enables defensible forensic inference for specific threat hypotheses. A tactic is "supportable" if at least one required network-forensic artifact (e.g., destination fan-out for Lateral Movement) is computable from the available evidence representation. This is distinct from detection accuracy: a detection system may achieve high accuracy, but if the required artifacts are not computable from the telemetry representation, the forensic inference cannot be defensibly supported for incident response purposes. This work assumes backbone passive monitoring (no payload, no endpoint context, no ground truth).

3. Materials and Methods

This work provides an audit procedure for evaluating how modern network monitoring practices affect the feasibility and depth of threat inference. We develop a reproducible audit procedure that identifies the limits of inference that arise when network evidence is progressively reduced through aggregation and abstraction.

Figure 1 illustrates the overall architecture and flow process of the audit framework. The framework takes three evidence representations (L0: packet-level, L1: flow-level, L2: time-aggregated) as input and produces a forensic coverage report that quantifies which threat hypotheses remain supportable at each abstraction level. The process consists of four

main stages: (1) Artifact Extraction, where observable network characteristics are computed from each evidence layer; (2) Artifact-to-Tactic Mapping, where literature-grounded associations link artifacts to ATT&CK tactic-level hypotheses; (3) D3FEND Applicability Analysis, where defensive technique categories are evaluated based on available evidence requirements; and (4) Coverage Metrics Computation, where inference coverage and defensive applicability are quantified to produce the final audit report.

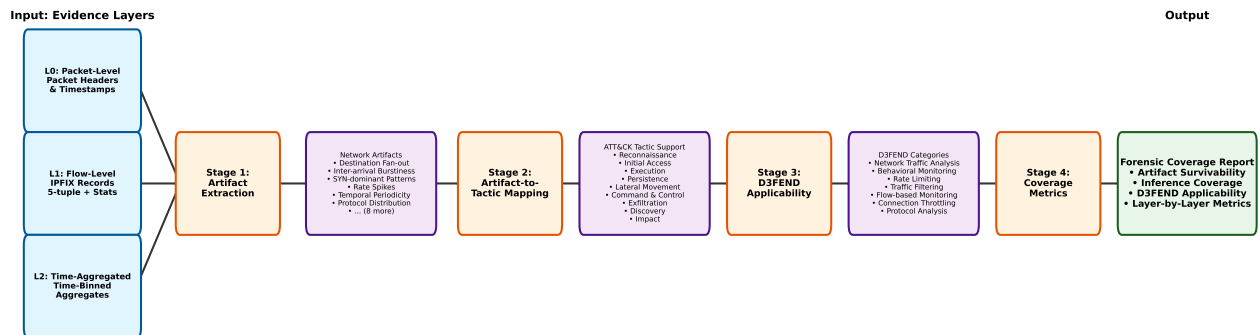


Figure 1. Architecture and flow process of the audit framework. The framework processes three evidence layers (L0: packet headers, L1: IPFIX flow records, L2: time-aggregated flows) through four main stages: (1) Artifact Extraction computes observable network characteristics from each layer; (2) Artifact-to-Tactic Mapping uses literature-grounded associations to link artifacts to ATT&CK tactic-level hypotheses; (3) D3FEND Applicability Analysis evaluates which defensive technique categories remain actionable given available evidence; (4) Coverage Metrics Computation produces quantitative measures (artifact survivability, inference coverage, D3FEND applicability) that form the forensic coverage report. The diagram highlights how evidence abstraction at each stage progressively constrains the set of supportable threat hypotheses and actionable defensive techniques.

To illustrate the core mechanism, consider a concrete example: The artifact “destination fan-out” (number of unique destinations per source) is required to support the ATT&CK Lateral Movement hypothesis. This artifact is computable from flow records (L1) but lost in time-aggregated counts (L2) where only bin-level aggregate statistics remain. Consequently, Lateral Movement becomes non-supportable at L2, regardless of how suspicious the aggregate statistics appear. This example demonstrates how evidence abstraction directly constrains which threat hypotheses can be defensibly supported.

Our methodology is structured around a principled inference chain:

$$\text{Observed Network Artifacts} \rightarrow \text{ATT\&CK Tactic} - \text{Level Support (Hypothesis Space)} \rightarrow \text{D3FEND Defensive Applicability}.$$

This chain enables structured reasoning when evidence is incomplete or ambiguous.

3.1. Threat Model and Forensic Assumptions

Our backbone network monitoring context assumes a passive forensic observer operating under realistic backbone monitoring constraints. The observer has access to packet headers captured at a transit network, without payload visibility, endpoint context, or ground-truth labels. This setting reflects ISP/carrier monitoring realities: retrospective analysis of backbone traffic, ISP-level monitoring, or post-incident investigations relying on archived network traces. This framing explicitly avoids assumptions common in prior network forensics work: we do not assume enterprise visibility, endpoint logs, or ground truth. This positioning is uncommon in forensic inference studies, which typically assume richer visibility, and makes the analysis directly relevant to Security Operations Center (SOC)/Incident Response (IR) methodology and operational security contexts. Figure 2

illustrates how evidence abstraction and inference chain collapse occur across network monitoring layers.

No assumptions are made about attacker identity, malware families, or campaign attribution. Consequently, our analysis deliberately excludes ATT&CK techniques that require host-level visibility, payload inspection, or semantic knowledge of application-layer content. Threat reasoning is restricted to high-level behavioral patterns observable at the network level, and we use ATT&CK strictly at the tactic level as a vocabulary for organizing hypothesis spaces, not as a classification taxonomy. When we refer to tactics such as Execution, we mean execution-related network manifestations (e.g., timing patterns consistent with scheduled task execution or script execution) observable from packet headers and flow records, not endpoint-level execution events that require host visibility.

ATT&CK Tactic Scope: For clarity and reproducibility, we explicitly specify which ATT&CK tactics are in-scope versus out-of-scope for our backbone network monitoring context:

In-Scope Tactics (Network-Observable): Our analysis includes the following ATT&CK tactics that can be observed through network-level behavioral patterns without requiring endpoint visibility:

- **Command and Control (C2):** Observable via network communication patterns, beaconing behavior, protocol anomalies, and temporal periodicity.
- **Discovery:** Observable via scanning patterns, protocol distribution anomalies, connection attempt patterns, and ICMP protocol share.
- **Exfiltration:** Observable via traffic volume anomalies (byte rate spikes), directional asymmetry, and unusual data transfer patterns.
- **Execution:** Observable via inter-packet timing patterns that indicate scheduled task execution or script execution (requires packet-level timing).
- **Impact:** Observable via traffic volume spikes, protocol anomalies, and rate-based anomalies.
- **Initial Access:** Observable via connection patterns, SYN-dominant connection bursts, and connection attempt anomalies.
- **Lateral Movement:** Observable via destination fan-out patterns, scanning behavior, and per-source connectivity anomalies.
- **Persistence:** Observable via entity-linked temporal periodicity, periodic communication patterns, and recurring beacon-like behavior.
- **Reconnaissance:** Observable via scanning patterns, protocol distribution imbalances, connection attempt patterns, and ICMP protocol share.

Out-of-Scope Tactics (Endpoint-Dependent): Our analysis excludes the following ATT&CK tactics that require endpoint visibility, payload inspection, or host-level context not available in backbone network traces:

- **Credential Access:** Requires endpoint visibility (password hashes, credential dumps, authentication logs) not available in backbone network traces.
- **Privilege Escalation:** Requires endpoint process/privilege visibility, system call monitoring, and host-level context not observable at the network level.
- **Defense Evasion:** Many techniques require endpoint context (process manipulation, file system changes, registry modifications) not observable in network traces.
- **Collection:** While some techniques are network-observable (data staging via network transfers), many require endpoint file system visibility and local data collection activities.

This explicit specification ensures that independent analysts can reproduce our analysis by clearly understanding which tactics are evaluated and which are excluded, and why.

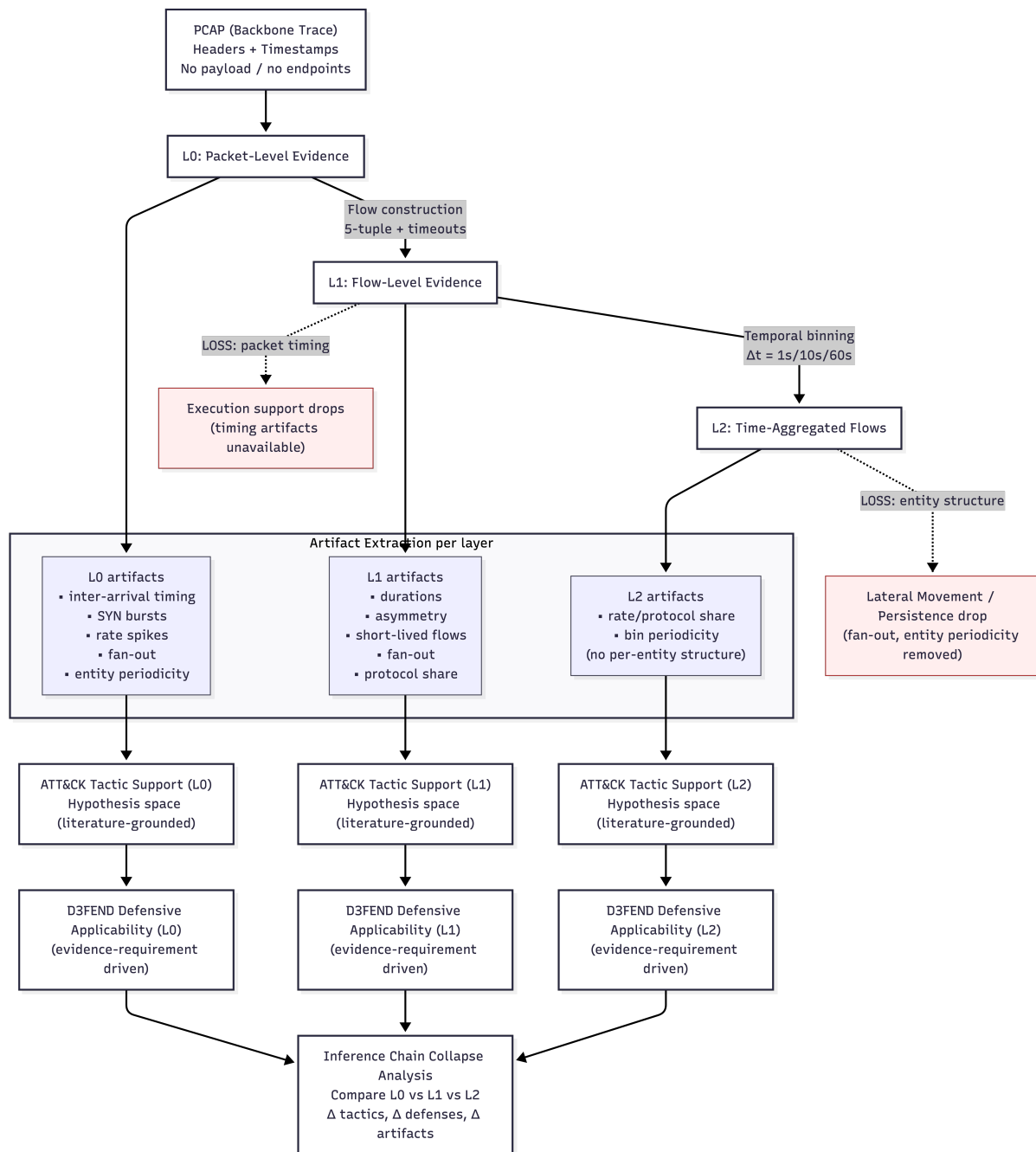


Figure 2. Evidence abstraction and inference chain collapse across network monitoring layers. Starting from backbone packet traces, network evidence is progressively transformed from packet-level (L0) to flow-level (L1) and time-aggregated (L2) representations. At each layer, only artifacts computable from the available evidence are extracted and used to support literature-grounded ATT&CK tactic-level hypothesis spaces. Defensive reasoning is performed using MITRE D3FEND based solely on evidence requirements, not defensive effectiveness. Artifact Loss Labels: L0 $\rightarrow$ L1 transition loses packet-timing artifacts (inter-arrival time burstiness, inter-packet timing patterns) but gains flow-specific artifacts (flow duration distribution, directional traffic asymmetry, short-lived flow patterns). L1 $\rightarrow$ L2 transition loses entity-structure artifacts (destination fan-out, entity-linked temporal periodicity) due to the removal of per-entity keys by temporal aggregation. Tactic Supportability Changes: Execution loses support at L1 (requires packet timing), while Lateral Movement and Persistence lose support at L2 (require entity structure). The figure highlights where abstraction induces loss of packet-timing and entity-structure artifacts, leading to the collapse of defensible tactic support and defensive applicability.

3.2. Evidence Representation Layers

Starting from raw packet capture (PCAP) data, we construct three increasingly abstract representations of network evidence:

1. Packet-Level Evidence: Individual packets with full header information and precise inter-packet timing.
2. Flow-Level Evidence: Bidirectional flows constructed using a standard 5-tuple with timeout-based aggregation.
3. Time-Aggregated Flow Evidence: Flow statistics further aggregated into fixed temporal bins (e.g., 10 s or 60 s intervals), emulating common monitoring and logging practices.

These representations allow us to model evidence loss as a controlled, stepwise process, enabling systematic analysis of how forensic visibility degrades as monitoring granularity decreases.

3.3. Network Artifact Extraction

From each evidence layer, we extract a set of network forensic artifacts. Artifacts are defined as observable traffic characteristics that may support forensic reasoning, independent of any specific detection algorithm. Examples include SYN-dominant connection patterns, destination fan-out, rate spikes, protocol imbalance, temporal periodicity, and directional asymmetry.

Importantly, artifacts are treated as observations, not as indicators of confirmed malicious activity. Their presence motivates further reasoning, but, by itself, it does not constitute evidence of compromise or attack.

Artifact Selection Criteria and Completeness

To enhance methodological transparency, we explicitly document the criteria used to select the 13 network-forensic artifacts and assess their completeness for coverage analysis.

Selection Criteria: The 13 artifacts were selected based on five criteria:

1. Network-observable: Artifacts must be computable from network traffic without requiring endpoint visibility, payload inspection, or host-level context. This criterion ensures compatibility with backbone monitoring constraints.
2. Literature-grounded: Artifacts must have established associations with threat behaviors documented in published network forensics literature. This criterion ensures that artifact-to-tactic mappings are defensible and reproducible.
3. Operationally definable: Artifacts must be defined as exact computable functions over network evidence representations (L0/L1/L2). This criterion ensures reproducibility and enables automated computation.
4. Forensically significant: Artifacts must support defensible reasoning about threat hypotheses, not just statistical anomalies. This criterion ensures that artifacts enable meaningful forensic inference.
5. Representation-sensitive: Artifacts must exhibit different computability across evidence layers (L0/L1/L2) to enable analysis of abstraction effects. This criterion ensures that the audit can evaluate how abstraction affects inference capability.

Completeness Assessment: The 13 artifacts are intended as a representative but not exhaustive basis for coverage analysis. The artifact catalog is designed to:

- Cover the major categories of network-forensic evidence (timing-based, structural, rate-based, connection patterns, protocol distribution).
- Enable evaluation of all network-observable ATT&CK tactics (9 tactics).
- Provide sufficient diversity to demonstrate selective and structural inference loss patterns.

- However, we acknowledge that additional artifacts could be included (e.g., DNS query patterns, TLS handshake features, packet size distributions, application-layer metadata).

Representative vs. Exhaustive: We clarify the following:

- The 13 artifacts are representative of the broader space of network-forensic artifacts.
- They are selected to ensure coverage of all major artifact categories and all network-observable tactics.
- They are not intended to be exhaustive, as the space of possible network-forensic artifacts is large and context-dependent.
- Future work could expand the catalog to include additional artifacts (e.g., application-layer metadata, encrypted traffic analysis features, domain-specific artifacts).

Coverage Justification: The 13 artifacts provide sufficient coverage for the audit framework's objectives because they:

- Enable evaluation of all 9 network-observable ATT&CK tactics.
- Span all major artifact categories (timing, structural, rate-based, connection, protocol).
- Demonstrate both selective loss (specific tactics) and structural loss (relational evidence).
- Enable evaluation of D3FEND defensive applicability across all relevant technique categories.

Future Expansion: The artifact catalog could be expanded to include the following:

- Domain-specific artifacts (e.g., DNS-based, TLS-based, application-layer).
- Context-dependent artifacts (e.g., enterprise-specific, cloud-specific, IoT-specific).
- Emerging artifact types (e.g., encrypted traffic analysis, ML-derived features, behavioral fingerprints).

This explicit discussion clarifies that the artifact catalog is representative rather than exhaustive, and provides justification for why the 13 artifacts provide sufficient coverage of the audit framework's objectives.

Table 2 provides a succinct summary of all 13 network-forensic artifacts, their computability across evidence layers (L0, L1, L2), and the important characteristics they capture. This summary table complements the detailed artifact catalog in Appendix A by providing practitioners with a quick reference for understanding which artifacts are available at each evidence layer and what network characteristics they capture.

Table 2. Summary of all 13 network-forensic artifacts, their computability across evidence layers, and key characteristics. This table provides a quick reference for practitioners to understand which artifacts are available at each evidence layer and what network characteristics they capture. Note: The distinction between “Temporal Periodicity (entity-linked)” and “Aggregate Temporal Periodicity” is critical: entity-linked periodicity (lost at L2) enables per-entity persistence inference, while aggregate periodicity (survives at L2) only supports bin-level pattern detection without entity attribution.

Artifact	Available In	Key Characteristics	Forensic Significance
Inter-arrival Time Burstiness	L0	Microsecond-precision inter-packet timing, burst patterns	Enables detection of scheduled task execution and script execution timing patterns
Inter-packet Timing Patterns	L0	Microsecond-precision inter-packet timing, temporal patterns	Enables detection of execution-related network manifestations and behavioral anomalies
SYN-Dominant Connection Bursts	L0, L1, L2	Connection initiation patterns, SYN flag dominance	Enables detection of scanning, connection attempts, and reconnaissance activities

Table 2. *Cont.*

Artifact	Available In	Key Characteristics	Forensic Significance
Packet Rate Spikes	L0, L1, L2	Per-time-bin packet count anomalies, volume spikes	Enables detection of execution, impact, and command and control activities via traffic volume anomalies
Byte Rate Spikes	L0, L1, L2	Per-time-bin byte count anomalies, data transfer spikes	Enables detection of exfiltration and impact activities via data volume anomalies
Destination Fan-out	L0, L1	Per-source destination connectivity patterns, scanning indicators	Enables detection of lateral movement and discovery activities via per-entity connectivity analysis
Temporal Periodicity (entity-linked)	L0, L1	Per-entity periodic communication patterns, beacon-like behavior	Enables detection of persistence and command and control activities via recurring communication patterns
Aggregate Temporal Periodicity	L0, L1, L2	Bin-level time-series periodicity, aggregate patterns	Enables detection of command and control activities via aggregate periodic patterns
Flow Duration Distribution	L1, L2	Flow lifetime statistics, connection duration patterns	Enables detection of execution and discovery activities via flow-based behavioral analysis
Directional Traffic Asymmetry	L1, L2	Bidirectional flow imbalance, upload/download ratios	Enables detection of exfiltration and lateral movement activities via directional traffic analysis
Short-Lived Flow Patterns	L1, L2	Flow duration anomalies, connection attempt patterns	Enables detection of reconnaissance and discovery activities via short-duration connection patterns
Protocol Distribution Imbalance	L0, L1, L2	Per-time-bin protocol share anomalies, protocol diversity	Enables detection of execution and discovery activities via protocol-level behavioral analysis
ICMP Protocol Share	L0, L1, L2	ICMP traffic proportion, protocol-specific patterns	Enables detection of discovery and reconnaissance activities via ICMP-based network probing

3.4. ATT&CK Tactic-Level Support (Literature-Grounded)

Observed artifacts are linked to tactic-level hypotheses using MITRE ATT&CK as a structured vocabulary, restricted to the tactic level. Technique-level identification requires a host context unavailable in backbone traces, so we restrict analysis to tactics as coarse-grained intent categories.

We use ATT&CK tactics as a vocabulary to organize a literature-grounded hypothesis space: an artifact provides support only for those tactics for which a published source describes a consistent network-visible behavioral relationship. We treat ATT&CK as a framework for organizing defensible reasoning, not as a taxonomy for classification. In all cases, the associations are non-exclusive and uncertainty-aware: multiple tactics may remain plausible for a given artifact, and absence of evidence is not treated as evidence of absence.

ATT&CK serves as a structured vocabulary for expressing what can reasonably be supported by available evidence, rather than as a taxonomy for detection or classification.

3.5. D3FEND-Based Defensive Reasoning

For each artifact and its associated tactic-level support relationships, we identify relevant defensive technique categories from the MITRE D3FEND framework. D3FEND is employed to reason about defensive applicability (which techniques remain actionable given available evidence) rather than defensive effectiveness or deployment status.

We map artifacts to D3FEND technique categories based on the observable evidence requirements of each technique group, identifying which defensive controls remain logically relevant given the surviving network evidence. The defensive technique categories referenced in our analysis (network traffic analysis, behavioral monitoring, rate limiting, traffic filtering, flow-based monitoring, connection throttling, and protocol analysis) represent our grouping of D3FEND techniques based on shared evidence requirements, aligned with, but not identical to, the official D3FEND taxonomy. These mappings are based on the evidence requirements documented in the D3FEND knowledge base [40]. No claims are made regarding the operational presence, performance, or evaluation of these controls.

Figure 3 illustrates the mapping process from artifacts to D3FEND defensive technique categories. The figure shows how observable artifacts (e.g., destination fan-out, rate spikes, temporal periodicity) enable specific defensive technique categories based on their evidence requirements. For example, destination fan-out enables Network Traffic Analysis by providing per-entity connectivity information required for traffic pattern analysis, while rate spikes enable Rate Limiting and Traffic Filtering by providing volume-anomaly information required for rate-based controls. This visualization clarifies how artifact availability at different evidence layers (L0, L1, L2) determines which defensive techniques remain applicable.

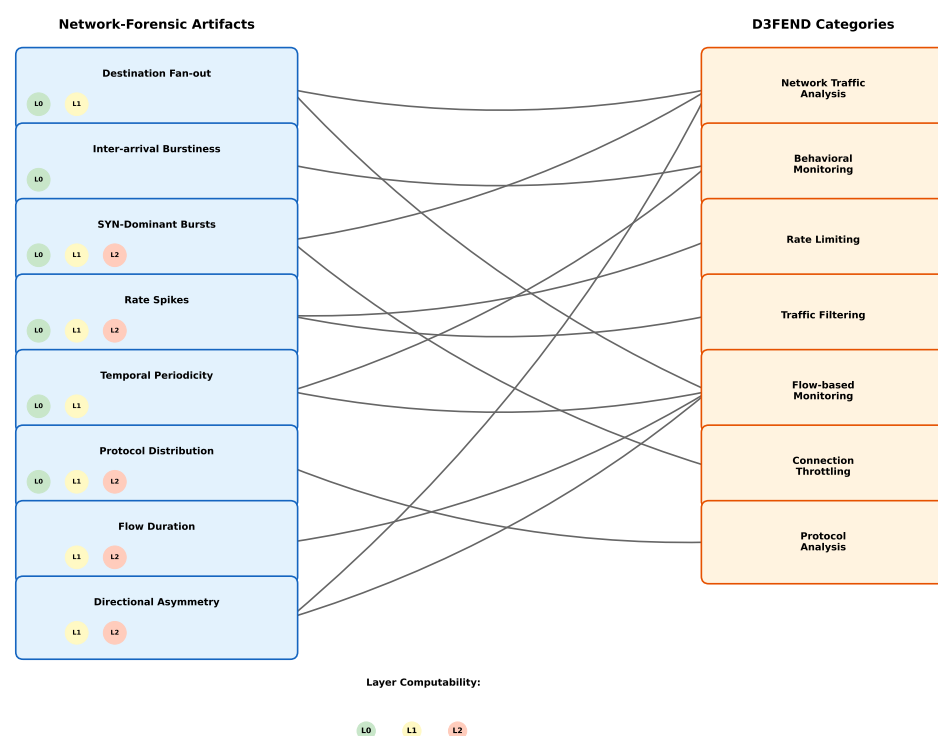


Figure 3. Mapping from network-forensic artifacts to D3FEND defensive technique categories. The figure illustrates how observable artifacts enable specific defensive techniques based on their evidence requirements. Artifacts are shown on the left, with their computability across evidence layers (L0, L1, L2) indicated. D3FEND categories are shown on the right, with arrows indicating which artifacts enable each category. This visualization clarifies how evidence abstraction affects defensive applicability: artifacts lost at higher abstraction levels (e.g., destination fan-out at L2) disable corresponding defensive techniques (e.g., Network Traffic Analysis for per-entity patterns).

3.6. Inference Chain Collapse Analysis

The final step of our approach analyzes how the inference chain degrades as evidence is reduced. By comparing artifact visibility and support relationships across packet-level, flow-level, and time-aggregated representations, we identify where defensible reasoning collapses from artifact-supported tactic hypotheses to non-specific or non-actionable observations.

This analysis allows us to characterize *evidence loss* not merely as a loss of data volume or resolution, but as a loss of inferential power within structured threat-reasoning frameworks. The result is a qualitative but systematic assessment of which forensic conclusions remain defensible under modern network monitoring constraints.

3.7. Fully Worked Example: Destination Fan-Out to Lateral Movement Mapping

To enhance methodological transparency and demonstrate consistency, we provide a complete worked example that traces the inference chain from artifact definition through literature grounding to the determination of tactic supportability.

Step 1: Artifact Definition. Destination fan-out is operationally defined as the number of unique destination IP addresses contacted by a source IP within a specified time window. This artifact captures per-source connectivity patterns that may indicate scanning or lateral movement activities.

Step 2: Evidence Layer Computability.

- L0 (Packet-level): Computable by parsing packet headers and counting unique destination IPs per source IP within time windows. Each packet header contains source and destination IP addresses, enabling direct computation.
- L1 (Flow-level): Computable by aggregating flow records and counting unique destination IPs per source IP. Flow records preserve source-destination pairs, enabling per-entity connectivity analysis.
- L2 (Time-aggregated): NOT computable because temporal aggregation removes per-entity source-destination pairs. Only aggregate counts per time bin remain (e.g., total flows, total bytes), losing the per-source structure required for fan-out computation.

Step 3: Literature Reference. Maghsoudlou et al. [10] describe scanning patterns and per-source connectivity analysis as indicators of lateral movement activities. The literature establishes that destination fan-out (a high number of unique destinations per source) is a network-visible behavioral pattern associated with lateral movement tactics, providing a literature-grounded association between the artifact and the tactic.

Step 4: Supported Tactic. *Lateral Movement* is supportable at L0 and L1 because destination fan-out is computable at these layers, enabling per-source connectivity analysis. When destination fan-out is observable, analysts can defensibly reason about whether observed traffic patterns are consistent with lateral movement hypotheses grounded in the literature.

Step 5: Non-Supported Tactic. *Lateral Movement* becomes non-supportable at L2 because destination fan-out is not computable from aggregated statistics. Even if aggregate patterns appear suspicious (e.g., high total flow counts, unusual protocol distribution), the forensic inference cannot be defensibly supported because the required artifact (per-source destination fan-out) is not available in the evidence representation. This demonstrates that artifact computability is a necessary condition for tactic supportability, independent of how suspicious aggregate statistics may appear.

This worked example demonstrates the complete inference chain: artifact definition → evidence layer computability → literature-grounded association → supported tactic determination → non-supported tactic identification. The methodology ensures consistency

by applying this structured process to all artifact-tactic pairs, with complete documentation in Appendix A and machine-readable decision rules in the Supplementary Materials.

3.8. Safeguards Against Subjective Attribution

A central methodological risk in forensic analysis of unlabeled network traffic is the introduction of subjective or non-reproducible attribution, particularly when higher-level threat frameworks such as MITRE ATT&CK are involved. We avoid performing tactic inference or defining new artifact-to-tactic rules. Instead, ATT&CK is used as a reference taxonomy to organize and evaluate the stability of forensic attributions that have already been established in prior literature.

3.8.1. Literature-Grounded Attribution

All associations between observed network artifacts and ATT&CK tactics are grounded exclusively in previously published sources, including official ATT&CK behavioral descriptions, detection guidance, and well-established network forensics literature. The analysis does not assert that a given artifact *implies* a specific adversarial tactic. Rather, it examines whether artifacts that have been *previously cited* as supporting tactic-level interpretation remain observable and defensible under different evidence representations. As such, attribution is treated as a documented association, not as a classification decision made by the authors.

3.8.2. Tactic-Level Restriction

To further reduce interpretive bias, the analysis is restricted to the ATT&CK tactic level. Technique-level attribution is avoided, as it would require assumptions about host context, payload visibility, or attacker intent that are not available in backbone traffic traces. Tactics are used solely as coarse-grained intent categories that allow comparison of support relationship stability across evidence representations.

3.8.3. No Ground Truth Assumption

This study does not assume the availability of ground truth labels, nor does it attempt to validate the correctness of any individual attribution. The objective is not to determine whether a specific tactic occurred, but to analyze how the *defensibility* of commonly cited forensic interpretations degrades as evidence is transformed from packet-level traces to flow-level and time-aggregated representations. Consequently, the analysis focuses on relative loss of defensible support relationships rather than absolute correctness.

3.8.4. Canonical Behavior Scenarios

When illustrative context is required, the analysis refers to canonical network behaviors (e.g., scanning, flooding, periodic communication) widely recognized in the literature and frequently used as reference cases in network forensics. These scenarios are not treated as confirmed attacks but as conceptual anchors that enable structured comparison of how their observable artifacts survive or collapse under evidence degradation.

3.8.5. Reproducibility and Transparency

All artifact definitions, evidence transformations, and attribution references are explicitly documented to ensure reproducibility. Because the methodology relies on published descriptions and observable properties rather than expert judgment or heuristic rule construction, independent analysts can replicate the analysis using the same data sources and reference materials.

To further enhance reproducibility and reduce interpretive subjectivity, we provide structured decision rules that encode explicit if-then criteria for determining artifact-to-

tactic supportability. These decision rules are based on evidence property requirements and literature-grounded associations, enabling independent validation of mapping decisions. Complete decision rules for all 13 artifacts with full citation lists are available in Appendix A and in machine-readable format (`decision_rules.json`) in the Supplementary Materials. When literature sources provide conflicting associations, we prioritize official ATT&CK documentation and explicitly document ambiguity in the decision rules.

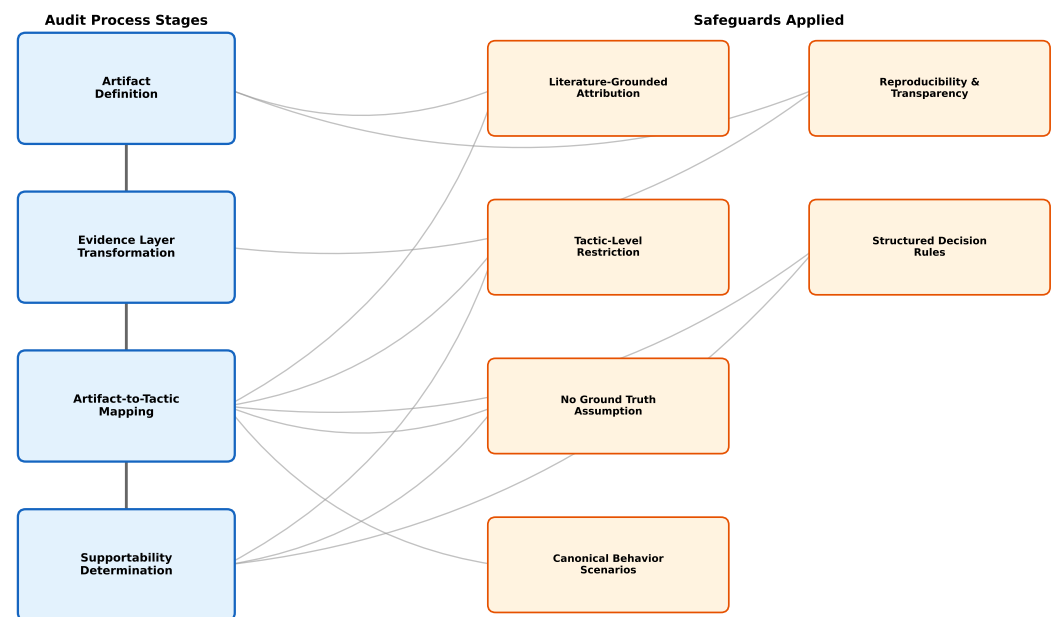
By constraining attribution to literature-backed associations, restricting analysis to coarse-grained intent categories, providing structured decision rules for reproducible mapping (documented in the Supplementary Materials), and reframing reasoning as a question of evidential stability rather than detection accuracy, the proposed methodology minimizes subjectivity while remaining faithful to the realities of forensic analysis under partial observability.

Table 3 provides a structured overview of the safeguards against subjective attribution, summarizing each approach and its role in ensuring methodological rigor. Figure 4 illustrates the flow of safeguards applied throughout the audit process, from artifact definition through tactic supportability determination, showing how each safeguard reduces subjectivity at different stages of the analysis.

Table 3. Summary of safeguards against subjective attribution in the audit framework. Each safeguard addresses a specific source of subjectivity and ensures methodological rigor.

Safeguard	Approach	Role in Reducing Subjectivity
Literature-Grounded Attribution	All artifact-to-tactic associations grounded exclusively in published sources (official ATT&CK descriptions, detection guidance, network forensics literature)	Eliminates author interpretation; uses only documented associations from prior literature
Tactic-Level Restriction	Analysis restricted to ATT&CK tactic level; technique-level attribution avoided	Reduces interpretive bias by avoiding assumptions about host context, payload, or attacker intent unavailable in backbone traces
No Ground Truth Assumption	Study does not assume ground truth labels or validate correctness of attributions	Focuses on defensibility of interpretations rather than absolute correctness, avoiding subjective validation judgments
Canonical Behavior Scenarios	Uses widely recognized network behaviors (scanning, flooding, periodic communication) as conceptual anchors	Provides objective reference points for comparison without requiring confirmed attack labels
Reproducibility and Transparency	All artifact definitions, evidence transformations, and attribution references explicitly documented	Enables independent validation; relies on observable properties rather than expert judgment
Structured Decision Rules	Explicit if-then criteria for determining artifact-to-tactic supportability (Table 5)	Provides formal, reproducible decision procedure that reduces interpretive judgment

Note: These safeguards work together to minimize subjectivity while maintaining fidelity to forensic analysis realities under partial observability.



Safeguards Against Subjective Attribution Flow

Figure 4. Flow diagram illustrating safeguards against subjective attribution applied throughout the audit process. The diagram shows how each safeguard (Literature-Grounded Attribution, Tactical-Level Restriction, No Ground Truth Assumption, Canonical Behavior Scenarios, Reproducibility and Transparency, Structured Decision Rules) is applied at different stages: artifact definition, evidence layer transformation, artifact-to-tactic mapping, and supportability determination. This visualization demonstrates how multiple safeguards work together to ensure methodological rigor and reduce subjectivity at each step of the analysis.

4. Results

We evaluate how monitoring abstraction affects the availability of network forensic artifacts and, consequently, the supportability of higher-level threat reasoning within the audit framework. Using three 15 min backbone traffic windows from MAWI DITL [8] (9 April 2025; 00:00, 06:00, 12:00), we compare three evidence representations: packet-level (L0), flow-level (L1), and time-aggregated flows (L2). Our evaluation assesses how evidence reduction constrains the space of supportable threat hypotheses from network observations, given the available evidence representation. This is a structural analysis of representational constraints, not a statistical measurement of traffic behavior. Accordingly, numerical values of artifacts are reported only to validate computability under realistic conditions, not to characterize traffic distributions or detect attacks. We evaluate representational survivability of literature-grounded support relationships, not correctness of adversary hypotheses.

4.1. Dataset and Experimental Setup

We evaluated three 15 min traffic windows from the MAWI DITL [8] backbone trace captured on 9 April 2025, at 00:00, 06:00, and 12:00. These windows were selected to provide temporal separation while keeping the pipeline computationally reproducible under realistic resource constraints.

Invariance by Design: Artifact computability is definitionally determined by the evidence representation (L0/L1/L2), not by traffic characteristics or time period length. For example, inter-packet timing patterns are either computable from packet-level traces

(L0) or not computable from flow records (L1) due to the structural loss of per-packet timestamps, regardless of traffic volume, composition, or time window duration. This structural property means that longer time windows (hours, days, weeks) or more varied windows (different days, different ISPs) would produce identical computability results for the same telemetry schema (L0/L1/L2). The audit framework evaluates structural properties of evidence representations, not statistical properties of traffic.

Empirical Validation: While artifact computability is invariant by design, real backbone traffic provides essential validation: (i) it confirms that artifact definitions are computable under realistic data volumes and formats (e.g., handling IPv4/IPv6, fragmented packets, and flow timeout boundaries), (ii) it reveals distributional properties of artifacts (e.g., prevalence of SYN-dominant patterns, typical fan-out ranges) that inform practical audit interpretation, and (iii) it demonstrates that the audit procedure remains stable across diverse traffic compositions (night, morning, noon windows), confirming representation-driven rather than traffic-driven effects. The consistent results across three diverse time windows (00:00 night traffic, 06:00 morning traffic, 12:00 noon traffic) demonstrate that artifact computability is representation-driven rather than traffic-driven. More varied windows would produce identical computability results for the same representation.

What Would Change: While artifact computability is invariant, the actual values of artifacts (e.g., specific fan-out counts, timing patterns, rate spike magnitudes) would vary with traffic composition and time period. However, the audit framework evaluates whether artifacts are computable (a structural property), not their specific values (a traffic-dependent property). This distinction is crucial: the audit answers “which threat hypotheses remain supportable?” not “what are the specific artifact values?” While MAWI is not representative of all backbone environments, the audit framework is agnostic to traffic source and applies to any telemetry pipeline exporting equivalent representations (packet headers, flow records, or time-aggregated statistics).

Software Tools and Configuration

For reproducibility, we document the complete software toolchain and configuration parameters. L0 packet extraction uses YAF [41] (Yet Another Flowmeter) version 2.14.0 to extract packet header fields and timestamps from PCAP files without payload access. YAF serves as both a flow meter and a packet header parser, reading packet headers directly from PCAP files. L1 flow generation uses YAF version 2.14.0 [41], configured with default flow timeout settings (active timeout: 60 s, idle timeout: 15 s) and 5-tuple flow key (source IP, destination IP, source port, destination port, protocol). YAF is invoked via command-line: `yaf -in <input.pcap> -out <output.yaf>`, which generates IPFIX-compliant flow records. YAF’s binary IPFIX output is converted to tabular format using `yafscii` (included with YAF distribution). L2 aggregation is performed using Python pandas (version 1.5.3) with fixed temporal bins (1 s, 10 s, 60 s). All artifact computation scripts are implemented in Python 3.10+ using pandas (1.5.3), numpy (1.24.3), and standard libraries. The complete pipeline, including artifact computation, survivability analysis, and metric generation, is available as Supplementary Material. Computational environment: macOS 14.x, Python 3.10, 16 GB RAM. Processing time per 15 min window: L0 extraction ~15 min, L1 flow generation ~45 min, L2 aggregation ~5 min, artifact computation ~10 min. These timings are reported for transparency and reproducibility only, not as performance benchmarks, and will vary depending on traffic volume, capture settings, and hardware configuration. Scalability considerations for extended periods and high-speed traffic are discussed in Section 6.7.4.

For each window, we evaluated three evidence representations:

- **L0 (Packet-level):** Packet header fields and timestamps extracted directly from Packet Capture (PCAP) files using YAF [41] to read packet headers without payload access

(MAWI trace capture length CapLen = 96 bytes). All L0 artifacts are computed only from captured headers and timestamps, with no access to payload fields.

- **L1 (Flow-level):** Bidirectional flow records exported by YAF [41] (Yet Another Flowmeter), a flow monitoring tool that generates IP Flow Information Export (IPFIX)-compliant flow records. We use *yafscii* to convert YAF's binary IPFIX output to tabular format. Flow definition is based on a quintuple (source IP, destination IP, source port, destination port, protocol) with timeout-based aggregation.
- **L2 (Time-aggregated):** L1 flows aggregated into fixed temporal bins (1 s, 10 s, 60 s).

For comparability across layers, L0 and L1 artifact computation uses 1-s binning (using packet timestamps at L0 and flow start times at L1). Protocol share at L1 is computed as the fraction of *flows* per protocol within each 1-s bin using the protocol field from YAF's IPFIX export schema. This temporal binning enables consistent computation of time-dependent artifacts such as protocol share and rate spikes across all three layers, though the L1 computation represents an approximation based on flow-level aggregation rather than direct packet counting.

The analysis processed gzip-compressed PCAP files of 4.6 GB, 2.7 GB, and 6.6 GB for the 00:00, 06:00, and 12:00 windows, respectively, totaling 13.9 GB of compressed network traffic (Flow record counts depend on the exporter's flow definition (5-tuple, timeout settings, and aggregation rules). The YAF export produced CSV files with 49,374,556, 47,600,000, and 52,600,000 rows, respectively (as counted from exported CSV files), where each row represents a flow record as defined by YAF's IPFIX export configuration. These counts reflect the actual number of flow records exported by YAF for each 15 min window).

We compute a catalog of 13 network-forensic artifacts. Each artifact is defined operationally (i.e., exact computable function over L0/L1/L2). Table 4 provides a compact excerpt of artifact-to-tactic mappings with representative citations; the complete artifact catalog with all literature-grounded tactic-support and D3FEND-applicability references is provided in Appendix A and available in the Supplementary Materials, where each association is accompanied by its source citation. The defensive technique categories referenced in our analysis (network traffic analysis, behavioral monitoring, rate limiting, traffic filtering, flow-based monitoring, connection throttling, and protocol analysis) represent our grouping of D3FEND techniques based on shared evidence requirements, aligned with the D3FEND knowledge base [40].

Table 4. Excerpt of artifact-to-tactic mappings (literature-grounded). Full catalog with all citations available in Supplementary Materials.

Artifact	Supports Tactics	Representative Citation
Inter-arrival Time Burstiness	Execution, Command and Control	Network timing patterns for behavioral inference [3]
SYN-Dominant Connection Bursts	Reconnaissance, Initial Access, Discovery	Encrypted traffic analysis and connection patterns [34]
Destination Fan-out	Lateral Movement	Scanning patterns and per-source connectivity [10]
Entity-linked Temporal Periodicity	Persistence	Periodic communication patterns in TLS traffic [12]
Protocol Distribution Imbalance	Reconnaissance, Discovery	Protocol evolution and distribution analysis [1]

Table 5 provides structured decision rules that encode explicit if-then criteria for determining artifact-to-tactic supportability based on evidence property requirements and literature-grounded associations.

Table 5. Structured decision rules for artifact-to-tactic supportability. Each rule provides explicit criteria for determining whether an artifact supports a tactic based on evidence properties and literature-grounded associations.

Artifact Category	Required Evidence Properties	Literature Criterion	Supportability Decision Rule	Uncertainty Handling
Timing-based (Inter-arrival Burstiness, Inter-packet Timing)	Per-packet timestamps with microsecond precision	ATT&CK Execution techniques include scheduled task execution, which requires inter-packet timing to distinguish from normal traffic [3]	IF artifact requires per-packet timestamps AND representation provides per-packet timestamps (L0), THEN artifact supports Execution inference; ELSE artifact does not support Execution inference (L1, L2)	None (consensus in literature)
Structural (Destination Fan-out)	Source-destination pairs per time window, per-entity keys preserved	Lateral Movement tactics involve scanning patterns observable via per-source destination fan-out [10]	IF representation preserves source-destination pairs per entity (L0, L1), THEN artifact supports Lateral Movement inference; ELSE artifact does not support Lateral Movement inference (L2)	Flow timeout effects may degrade precision at L1—see Supplementary
Entity-linked Temporal (Periodicity)	Per-entity temporal patterns, entity identifiers preserved	Persistence tactics involve periodic beacon-like communication patterns observable via entity-linked periodicity [12]	IF representation preserves per-entity temporal patterns (L0, L1), THEN artifact supports Persistence inference; ELSE artifact does not support Persistence inference (L2)	Flow timeout effects may degrade precision at L1—see Supplementary
Rate-based (Packet/Byte Rate Spikes)	Flow or packet counts per time bin	Execution, Impact, and Exfiltration tactics involve traffic volume anomalies observable via rate spikes [34]	IF representation provides flow or packet counts per time bin (L0, L1, L2), THEN artifact supports Execution/Impact/Exfiltration inference; ELSE artifact does not support rate-based inference	None (consensus in literature)
Connection Pattern (SYN-Dominant Bursts)	Connection attempt patterns, SYN flags or flow initiation patterns	Reconnaissance, Initial Access, and Discovery tactics involve connection scanning observable via SYN-dominant patterns [34]	IF representation provides connection initiation information (SYN flags at L0, flow start indicators at L1/L2), THEN artifact supports Reconnaissance/Initial Access/Discovery inference; ELSE artifact does not support connection-based inference	None (consensus in literature)

Note: Complete decision rules for all 13 artifacts with full citation lists are available in Appendix A and in machine-readable format (`decision_rules.json`) in the Supplementary Materials. When literature sources provide conflicting associations, we prioritize official ATT&CK documentation and explicitly document ambiguity in the uncertainty handling column.

4.2. Artifact Survivability Across Layers

Table 6 reports whether each artifact is computable in each evidence layer. Timing-sensitive artifacts (inter-arrival burstiness, inter-packet timing) are available only in packet-level evidence (L0), while flow-specific artifacts (duration distribution, directional asymmetry) emerge at L1 and persist at L2. Protocol distribution imbalance and ICMP protocol share are defined as *protocol share per time bin* (the fraction of flows or packets per protocol within each temporal bin), making them computable at all three layers. ICMP share is included because ICMP traffic patterns can indicate reconnaissance or network discovery activities, and protocol-level statistics remain observable even under aggregation.

Table 6. Artifact computability across evidence layers. A checkmark indicates the artifact can be computed from that representation under our pipeline.

Artifact	L0	L1	L2
Inter-arrival Time Burstiness	✓	×	×
Inter-packet Timing Patterns	✓	×	×
SYN-Dominant Connection Bursts	✓	✓	✓
Packet Rate Spikes	✓	✓	✓
Byte Rate Spikes	✓	✓	✓
Destination Fan-out	✓	✓	×
Temporal Periodicity (entity-linked)	✓	✓	×
Aggregate Temporal Periodicity (bin-level)	✓	✓	✓
Flow Duration Distribution	×	✓	✓
Directional Traffic Asymmetry	×	✓	✓
Short-Lived Flow Patterns	×	✓	✓
Protocol Distribution Imbalance	✓	✓	✓
ICMP Protocol Share	✓	✓	✓

The table reveals three distinct artifact categories: (1) packet-level timing artifacts (inter-arrival burstiness, inter-packet timing) that are completely lost in flow-based representations, (2) flow-specific artifacts (duration distribution, directional asymmetry, short-lived flows) that emerge only at L1 and persist at L2, and (3) artifacts that remain stable across all layers (rate spikes, SYN patterns, protocol statistics). The loss of destination fan-out at L2 reflects a modeling decision that aligns with common operational practice: coarse telemetry exports aggregate counts and rates per time bin but do not preserve per-entity adjacency structures (e.g., $\text{source} \rightarrow \text{set}(\text{destinations})$). This design choice models realistic monitoring abstractions where storage and processing constraints favor aggregate statistics over per-entity keyed structures. Protocol-level statistics remain computable at all layers because of their per-time-bin definition, which aggregates flows per protocol within each 1-s bin.

Clarification on periodicity and Persistence. While *aggregate temporal periodicity* (bin-level time-series periodicity, e.g., periodic variation in total flow counts) remains computable at L2, the *entity-linked periodicity* required to support a *Persistence* hypothesis (e.g., a specific host exhibiting recurring beacon-like communication) is not computable when L2 aggregation discards per-entity keys (source/destination identifiers). Therefore, Persistence loses defensible support at L2 despite the continued availability of aggregate periodicity.

4.3. Inference Chain Collapse

We characterize reasoning degradation using two structural metrics: *inference coverage* ($|T_L|$), the number of ATT&CK tactics supported by at least one computable artifact in layer L (based on literature-grounded artifact-to-tactic associations), and *defensive applicability* ($|D_L|$), the number of D3FEND technique categories whose required observable inputs

are available in layer L . These metrics are *structural counts* that assess representational constraints, not statistical measurements of traffic behavior. We use *inference chain collapse* to denote the overall phenomenon, and *degradation* to refer to the stepwise loss of inferential power across $L0 \rightarrow L1 \rightarrow L2$.

Table 6 establishes which artifacts are computable at each layer. Table 7 shows how artifact loss translates to tactic-level inference loss: the *Execution* tactic loses support at L1 because its supporting artifacts (inter-arrival time burstiness, inter-packet timing) require packet-level timing information that is not computable from flow records. The *Lateral Movement* and *Persistence* tactics lose support at L2 because their supporting artifacts (destination fan-out, entity-linked temporal periodicity) require per-entity structural information that is removed by temporal aggregation. Table 8 shows how artifact loss affects defensive applicability: *Anomaly Detection* becomes non-applicable at L2 because it requires per-entity structural information, while *Flow-based Monitoring* becomes applicable at L1 once flow records are available.

The audit reveals that evidence abstraction reduces inference coverage from 9 (L0) to 7 (L2) tactics, while D3FEND applicability increases from 7 (L0) to 8 (L1) and then decreases to 7 (L2). The increase in available artifacts at L1 ($10 \rightarrow 11$) reflects the emergence of flow-specific properties (e.g., flow duration distribution, directional asymmetry, short-lived flow patterns) that cannot be defined at the packet level. Despite this quantitative increase, inference coverage decreases monotonically ($9 \rightarrow 8 \rightarrow 7$), demonstrating that *artifact quantity does not directly translate to inferential discriminability*. Detailed interpretation of these findings, including formal definitions and structured decision procedures, is provided in Section 6.

Table 7. ATT&CK tactic support availability across evidence layers.

Tactic	L0	L1	L2
Command and Control	Yes	Yes	Yes
Discovery	Yes	Yes	Yes
Execution	Yes	No	No
Exfiltration	Yes	Yes	Yes
Impact	Yes	Yes	Yes
Initial Access	Yes	Yes	Yes
Lateral Movement	Yes	Yes	No
Persistence	Yes	Yes	No
Reconnaissance	Yes	Yes	Yes

Table 8. D3FEND-aligned defensive technique category applicability across evidence layers. Categories represent our grouping of D3FEND techniques based on shared evidence requirements.

D3FEND Category	L0	L1	L2
Anomaly Detection	Yes	Yes	No
Behavioral Monitoring	Yes	Yes	Yes
Connection Throttling	Yes	Yes	Yes
Flow-based Monitoring	No	Yes	Yes
Network Traffic Analysis	Yes	Yes	Yes
Protocol Analysis	Yes	Yes	Yes
Rate Limiting	Yes	Yes	Yes
Traffic Filtering	Yes	Yes	Yes

Table 8 shows D3FEND defensive technique categories applicable at each layer. The increase in applicability at L1 ($7 \rightarrow 8$) is explained by the emergence of *Flow-based Monitoring*, which becomes enabled at L1 (requiring flow records), while *Anomaly Detection* is

disabled at L2 (requires per-entity structural information). This shift represents a transformation from behavioral monitoring (requiring fine-grained timing) to rate/flow-based controls (requiring aggregate statistics), reflecting the operational constraints of coarse telemetry systems.

4.4. Stability Across Time Windows

The audit procedure produces consistent collapse patterns across all three time windows (00:00, 06:00, 12:00), with identical metrics at each layer: 10/11/9 artifacts, 9/8/7 tactics, and 7/8/7 D3FEND categories. This invariance indicates that evidence-loss effects are structural properties of monitoring abstractions rather than artifacts of specific traffic characteristics. While the underlying artifact values (e.g., rates and protocol proportions) vary across windows, the computability pattern and the resulting coverage/applicability metrics remain invariant, indicating a representation-driven rather than traffic-driven effect. This consistency across diverse time periods (night, morning, noon) indicates that the audit framework produces stable results across the evaluated traffic conditions, supporting its use as an evaluation tool. Extended visualizations demonstrating this invariance across all three windows are provided in Appendix B.

Figure 5 provides a visual synthesis of the key collapse metrics for practitioners, combining artifact survivability, inference coverage, and D3FEND applicability into a single view to emphasize their non-monotonic and divergent behavior. This synthesis highlights that evidence abstraction does not produce uniform degradation: artifact count increases at L1 while inference coverage decreases, and D3FEND applicability follows a distinct trajectory from both metrics.

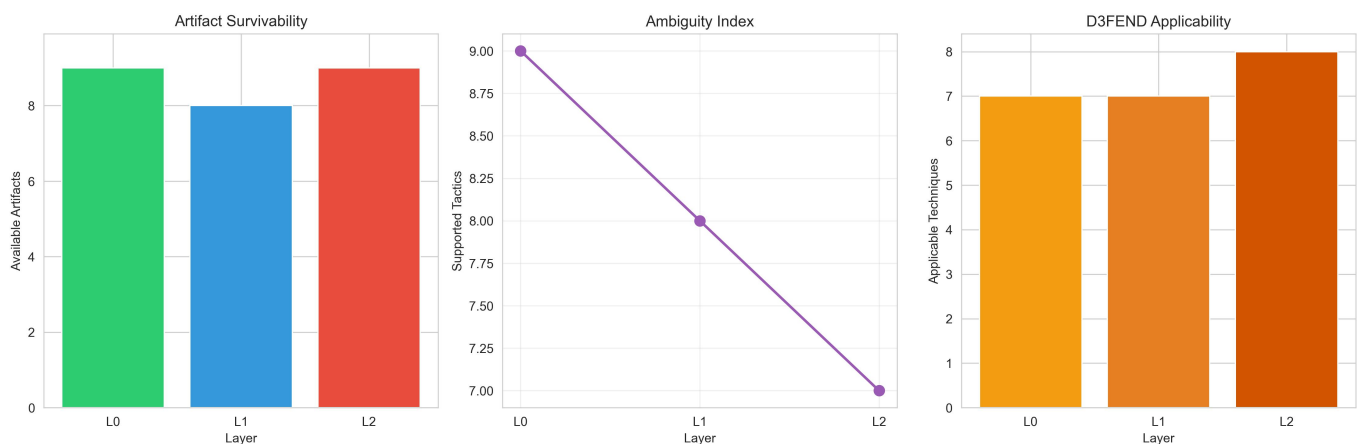


Figure 5. Visual synthesis of collapse metrics across evidence layers (included for practitioner reference). The figure combines artifact survivability, inference coverage, and D3FEND applicability to emphasize their non-monotonic and divergent behavior. The visualization highlights that evidence abstraction does not produce uniform degradation: artifact count increases at L1 (10 → 11) while inference coverage decreases monotonically (9 → 8 → 7), and D3FEND applicability increases at L1 (7 → 8) then decreases at L2 (8 → 7), reflecting a transformation from behavioral monitoring to flow-based controls.

5. Implications for Practice

The audit results indicate that evidence abstraction creates *selective blind spots* rather than uniform degradation. The 22% reduction in inference coverage (nine → seven tactics) constrains the space of supportable tactic hypotheses. This section translates these findings into actionable guidance for three key audiences.

5.1. For Network Architects

When evaluating a telemetry system, architects should ask: (1) Does it preserve source-destination pairs per time bin (required for *Lateral Movement* analysis via destination fan-out)? (2) Can we derive inter-packet timing from the exported data (required for *Execution* analysis)? (3) Are per-entity temporal patterns preserved, or only aggregate statistics (required for *Persistence* analysis)? The audit procedure provides a systematic method to answer these questions: given a telemetry schema, it identifies which artifacts remain computable and, consequently, which threat hypotheses can be supported.

For IPFIX extensions and programmable data plane telemetry, architects should prioritize preserving artifacts that enable high-value threat hypotheses. Our results suggest that preserving source-destination pairs (enabling destination fan-out) and inter-packet timing (enabling execution inference) should be prioritized over aggregate-only statistics. For example, if a system exports only time-aggregated flow counts per protocol (L2), destination fan-out is not computable, and *Lateral Movement* hypotheses cannot be supported regardless of how suspicious aggregate statistics appear. Architects should document not only what data is collected but also which forensic inference classes remain supportable under the audit framework.

5.2. For Incident Responders

If a playbook for investigating lateral movement requires analyzing destination fan-out, but archived data is time-aggregated (L2), that playbook step cannot be executed. Playbooks must be aligned with evidence availability. Specifically, if an organization relies on L2 telemetry (time-aggregated flows), claims that observed patterns support *Execution* (requires packet timing), *Lateral Movement* (requires destination fan-out), or *Persistence* (requires temporal periodicity with per-entity structure) hypotheses cannot be supported when only aggregated telemetry is available.

The defensive posture shifts from behavioral monitoring (requiring fine-grained timing) to rate/flow-based controls (requiring aggregate statistics). Incident response procedures must reflect these constraints: playbooks that require evidence types no longer available in archived telemetry cannot be executed given the available evidence. Organizations should align defensive strategies with monitoring capabilities rather than assuming that “more data” always improves security posture.

5.3. For Tool Developers

Tools could integrate this audit by generating a “forensic coverage report” alongside system deployment. Such a report would map the telemetry schema to supportable ATT&CK tactics and applicable D3FEND categories, enabling operators to understand the forensic capabilities of their monitoring infrastructure before incidents occur. The audit procedure can be automated: given a telemetry schema specification (e.g., an IPFIX template or a P4 telemetry export format), it outputs which artifacts are computable, which tactics remain supportable, and which defensive techniques are applicable. This enables proactive telemetry design: operators can evaluate multiple schema options and select configurations that preserve required forensic capabilities while meeting bandwidth and storage constraints. Integration with SDN controllers could enable dynamic telemetry adjustment: when threat intelligence indicates increased risk of lateral movement, the controller could temporarily increase export granularity to preserve destination fan-out artifacts.

In summary, network engineers can use this audit framework to make concrete design decisions: when configuring IPFIX exporters, they can determine which fields (e.g., `flowStartMilliseconds`, source-destination pairs) must be preserved to support specific threat hypotheses; when designing P4 telemetry pipelines, they can identify the

minimal export schema that maintains required forensic coverage; and when evaluating monitoring architectures, they can assess trade-offs between scalability and inferential capability before deployment.

6. Discussion

The audit results reveal a *non-monotonic transformation* of defensive capabilities under evidence abstraction: D3FEND applicability increases at L1 ($7 \rightarrow 8$) before decreasing at L2 ($8 \rightarrow 7$), while inference coverage decreases monotonically ($9 \rightarrow 8 \rightarrow 7$). This non-monotonicity has important implications for network monitoring architecture design. Rather than viewing abstraction as uniformly degrading forensic capability, network architects should recognize that different abstraction levels enable different classes of defensive reasoning.

6.1. Formal Framework and Decision Procedures

To reduce subjectivity and enhance reproducibility, we provide a formal definition of tactic supportability and structured decision procedures. This formalization ensures that supportability determinations are reproducible and based on explicit criteria rather than subjective interpretation.

6.1.1. Formal Definition of Tactic Supportability

Definition 1 (Tactic Supportability). *A tactic T is supportable at evidence layer L if and only if*

$$\exists A \in \mathcal{A}_L : \text{assoc}(A, T) \wedge \text{computable}(A, L) \quad (1)$$

where

- $\mathcal{A}_L$ is the set of artifacts computable at layer L .
- $\text{assoc}(A, T)$ denotes a literature-grounded association between artifact A and tactic T (documented with citations in Appendix A).
- $\text{computable}(A, L)$ indicates that artifact A is computable from evidence layer L (determined structurally, not empirically).

This formal definition provides explicit necessary and sufficient conditions: a tactic is supportable if and only if (1) at least one artifact is computable at the layer, and (2) that artifact has a literature-grounded association with the tactic.

Formal Inference Coverage Metric: We formalize the inference coverage metric as:

$$\text{Coverage}(L) = |\{T \in \mathcal{T} : \text{supportable}(T, L)\}| \quad (2)$$

where $\mathcal{T}$ is the set of network-observable ATT&CK tactics, and $\text{supportable}(T, L)$ is the binary supportability determination based on the definition above.

6.1.2. Structured Decision Procedure

To ensure consistency and reduce subjectivity, we apply the following structured decision procedure for each tactic T and layer L :

1. **Step 1: Artifact Computability Check:** Identify all artifacts A that are computable at layer L (structural determination based on evidence representation, no interpretation required).
2. **Step 2: Literature Association Check:** For each computable artifact A , check literature for established associations with tactic T (documented in Appendix A with complete citations).

3. **Step 3: Supportability Determination:** Apply the formal rule: tactic T is supportable if $\exists A$ such that both $\text{assoc}(A, T)$ and $\text{computable}(A, L)$ hold.
4. **Step 4: Ambiguity Documentation:** If multiple sources provide conflicting associations, document the ambiguity explicitly and prioritize official ATT&CK documentation.
5. **Step 5: Discrepancy Documentation:** Document any discrepancies between the decision rule application and expected outcomes, including cases where literature sources provide conflicting associations or where artifact computability may be context-dependent. This documentation enables independent validation and identifies areas requiring further clarification.

This structured procedure ensures that supportability determinations are reproducible and based on explicit criteria rather than subjective interpretation. All decision rules are encoded in machine-readable format (`decision_rules.json`) in the Supplementary Materials, enabling automated validation and independent reproduction.

6.2. Detailed Interpretation of Inference Chain Collapse

While individual losses may appear intuitive in isolation (e.g., flows remove packet timing, aggregation removes entity structure), their compound effect on tactic-level inference and defensive applicability has not been systematically evaluated. The audit reveals that evidence abstraction reduces inference coverage from 9 (L0) to 7 (L2) tactics, while D3FEND applicability increases from 7 (L0) to 8 (L1) and then decreases to 7 (L2), reflecting a shift from entity-aware anomaly reasoning to flow-based defensive techniques under aggregation.

The increase in available artifacts at L1 ($10 \rightarrow 11$) reflects the emergence of flow-specific properties (e.g., flow duration distribution, directional asymmetry, short-lived flow patterns) that are not definable at the packet level, partially compensating for the loss of fine-grained timing information. Despite this quantitative increase, inference coverage decreases monotonically ($9 \rightarrow 8 \rightarrow 7$), demonstrating that *artifact quantity does not directly translate to inferential discriminability*. The increase in D3FEND applicability at L1 ($7 \rightarrow 8$) is explained by the emergence of *Flow-based Monitoring*, which becomes applicable once flow records are available. The decrease at L2 ($8 \rightarrow 7$) reflects the loss of *Anomaly Detection*, which requires per-entity structural information that is removed by temporal aggregation. This shift represents a transformation from packet-level behavioral monitoring (requiring fine-grained timing) to flow-based defensive techniques (requiring aggregate flow statistics).

6.3. Scenario-Based Demonstration: Port-Scan-like Behavior

To concretely demonstrate how evidence abstraction creates ambiguity, we trace a canonical network behavior—port-scan-like patterns characterized by high destination fan-out, SYN-dominant connections, and short-lived flows—through each evidence layer. For the scenario trace, we selected a short sub-interval within the 15 min window exhibiting an extreme tail in (i) per-source destination fan-out and (ii) SYN-dominant, short-lived connection patterns, as measured by our artifact definitions; this selection is used solely to illustrate representation-driven evidence loss and does not constitute attack attribution. Figure 6 visualizes the 15 min window with the selected sub-interval highlighted, showing the same time period across L0, L1, and L2 representations.

At L0 (packet-level), a responder can observe fine-grained timing patterns (inter-packet intervals, burst dynamics) and flag-level detail (SYN ratios), enabling support for *Reconnaissance*, *Discovery*, and *Initial Access* tactics with high evidential strength. At L1 (flow-level), destination fan-out becomes directly computable per source, preserving the ability to support *Lateral Movement* inference while losing packet-level timing texture. At L2 (time-aggregated), destination fan-out is no longer computable because temporal

aggregation removes per-source keyed structure; only bin-level aggregate counts (total flows, rates) remain. This collapse eliminates support for *Lateral Movement* while preserving support for *Reconnaissance* and *Discovery* via SYN patterns and protocol statistics.



Figure 6. Trace of port-scan-like behavior across evidence layers. L0 shows packet-level rate spikes and SYN patterns; L1 shows per-source fan-out (visible); L2 shows only aggregate counts (fan-out not computable).

This scenario directly illustrates the logical progression from Table 6 to Table 7: the loss of destination fan-out at L2 (Table 6) eliminates support for *Lateral Movement* (Table 7), demonstrating that the “9 → 7 tactics” reduction is not uniform. Some tactics (e.g., *Lateral Movement*) become completely non-supportable at L2, while others (e.g., *Reconnaissance*) remain supportable but with reduced evidential strength due to loss of structural detail.

6.4. Comparative Analysis: Framework Positioning and Trade-Offs

To address concerns about comparative evaluation, we provide a systematic conceptual comparison of our audit framework with related approaches, highlighting advantages, disadvantages, and appropriate use cases. We note that traditional experimental comparisons (e.g., ROC-AUC, precision, recall) are not methodologically appropriate here because our framework evaluates representational limits of inference (a structural property), not

detection performance (a measurable accuracy metric). Detection frameworks and audit frameworks serve fundamentally different purposes and cannot be meaningfully compared using shared experimental metrics. Instead, we provide a qualitative comparison based on framework characteristics, objectives, and use cases. Table 9 compares our framework against representative works across key dimensions: evaluation objective, methodology, output, and operational applicability.

Table 9. Comparative analysis of the proposed audit framework against related approaches. The comparison highlights fundamental differences in objectives, methodologies, and outputs, demonstrating that each framework serves distinct purposes in network security and monitoring.

Aspect	Detection/Classification Frameworks	Telemetry System Frameworks	This Work (Audit Framework)
Primary Objective	Maximize detection accuracy, minimize false positives/negatives	Optimize telemetry export efficiency, reduce overhead	Evaluate which threat hypotheses remain supportable under evidence abstraction
Evaluation Metric	ROC-AUC, precision, recall, F1-score	Throughput, memory usage, export bandwidth	Artifact computability, inference coverage, D3FEND applicability
Methodology	Supervised/unsupervised learning, feature engineering, model training	Resource optimization, sampling strategies, data structure design	Structural analysis of representational constraints, literature-grounded artifact-to-tactic mapping
Output	Attack labels, confidence scores, alerts	Optimized telemetry configurations, performance metrics	Forensic coverage report (which tactics/defenses remain supportable)
Requires Ground Truth	Yes (for training/validation)	No (performance-focused)	No (structural analysis)
Use Case	Real-time threat detection, incident response	Network monitoring architecture design, resource-constrained environments	Design-time telemetry evaluation, forensic readiness planning
Advantages	High detection accuracy when trained on labeled data, real-time operation	Scalable, resource-efficient, production-ready	Independent of detection algorithms, reveals fundamental representational limits, design-time guidance
Limitations	Requires labeled datasets, algorithm-dependent, may not reveal why inference fails	Does not evaluate forensic coverage, focuses on efficiency, not inference supportability	Does not provide detection capabilities, requires manual interpretation of coverage reports

Note: This comparison demonstrates that frameworks serve complementary rather than competing roles. Detection frameworks optimize for accuracy, telemetry systems optimize for efficiency, while our audit framework evaluates representational limits of inference.

6.4.1. Comparison with Detection-Focused Frameworks

Detection frameworks (e.g., [6,7]) evaluate *how accurately* attacks can be identified across different data representations. These frameworks measure performance metrics (ROC-AUC, precision, recall) and optimize for detection effectiveness. In contrast, our audit framework evaluates *which threat hypotheses* can be logically supported given available evidence, independent of any specific detection algorithm.

Advantages of our approach:

- **Algorithm-agnostic:** Results hold regardless of detection method (ML, rule-based, statistical).
- **Reveals fundamental limits:** Identifies cases where inference is impossible due to evidence loss, not algorithm limitations.
- **No ground truth required:** Structural analysis does not depend on labeled attack datasets.
- **Design-time guidance:** Helps architects choose telemetry configurations before deployment.

Limitations relative to detection frameworks:

- **No detection capability:** Does not identify attacks or provide alerts.
- **No performance metrics:** Does not measure accuracy, false positive rates, or detection latency.
- **Requires interpretation:** Coverage reports must be manually analyzed to inform decisions.

When to use each: Detection frameworks are appropriate for operational security systems requiring real-time threat identification. Our audit framework is appropriate for design-time evaluation, forensic readiness planning, and understanding fundamental inference boundaries before deploying detection systems.

6.4.2. Comparison with Telemetry System Frameworks

Telemetry system frameworks (e.g., [14,15,32]) optimize monitoring architectures for efficiency, throughput, and resource utilization. These frameworks focus on *how to export* telemetry data efficiently, while our framework evaluates *what can be inferred* from exported data.

Advantages of our approach:

- **Forensic coverage evaluation:** Quantifies which threat hypotheses remain supportable, not just export efficiency.
- **Evidence-centric analysis:** Reveals how abstraction affects inference capabilities, not just performance.
- **Design guidance:** Provides criteria for choosing between telemetry configurations based on forensic requirements.

Limitations relative to telemetry frameworks:

- **No performance optimization:** Does not address throughput, memory, or bandwidth constraints.
- **No operational deployment:** Provides design-time analysis, not runtime telemetry export.
- **Static evaluation:** Assumes fixed telemetry configurations, does not handle dynamic adaptation.

When to use each: Telemetry frameworks are essential for designing scalable monitoring systems under resource constraints. Our audit framework complements these by evaluating whether efficient telemetry configurations preserve required forensic coverage, enabling informed trade-off decisions.

6.4.3. Comparison with ATT&CK/D3FEND Modeling Frameworks

ATT&CK/D3FEND modeling frameworks (e.g., [38,39]) use these vocabularies for strategic threat/defense modeling, often with rich visibility (endpoint logs, payload access, domain context). Our framework uses ATT&CK/D3FEND under backbone-only constraints to evaluate supportability.

Advantages of our approach:

- **Backbone-constrained:** Explicitly designed for partial observability (no payload, no endpoint context).
- **Supportability focus:** Evaluates which hypotheses remain defensible, not strategic modeling.
- **Representation-driven:** Links evidence abstraction directly to inference limits.

Limitations relative to modeling frameworks:

- **Narrower scope:** Restricted to network-observable tactics, excludes endpoint-dependent behaviors.
- **No strategic recommendations:** Does not provide game-theoretic or optimization-based defense strategies.
- **Tactic-level only:** Avoids technique-level attribution that requires richer context.

When to use each: Modeling frameworks are appropriate for enterprise environments with rich visibility and strategic defense planning. Our framework is appropriate for ISP/backbone monitoring where visibility is constrained, and the focus is on understanding inference boundaries.

6.4.4. Synthesis: Complementary Roles

This comparative analysis demonstrates that frameworks serve *complementary rather than competing* roles. Detection frameworks optimize for accuracy, telemetry systems optimize for efficiency, and modeling frameworks optimize for strategic defense. Our audit framework fills a distinct gap by evaluating *representational limits of inference* under evidence abstraction.

The audit framework is most valuable when used *before* deploying detection systems or telemetry architectures, helping architects understand which threat hypotheses will remain supportable under chosen configurations. It does not replace detection systems or telemetry optimizers; rather, it provides a prerequisite analysis that informs their design and configuration.

6.5. Design Principles for Hybrid Monitoring Systems

The increase in D3FEND applicability at L1 reflects the emergence of *Flow-based Monitoring*, which becomes actionable once flow records are available. This suggests that hybrid monitoring systems—combining selective packet capture with flow export—may optimize forensic coverage while maintaining scalability. For example, an SDN controller might dynamically adjust telemetry granularity: maintaining flow records for broad coverage while selectively enabling packet-level capture for specific flows exhibiting suspicious patterns (e.g., high destination fan-out). Our artifact catalog provides a principled basis for such adaptive telemetry: artifacts that require packet timing (e.g., inter-arrival burstiness) would trigger selective packet capture, while artifacts computable from flows (e.g., destination fan-out) would rely on flow export.

6.6. Implications for Programmable Data Plane Telemetry

Next-generation programmable data planes (e.g., the P4 programming language and the extended Berkeley Packet Filter (eBPF)) enable fine-grained control over telemetry export. Our artifact catalog directly informs what to measure: preserving source-destination pairs per time bin enables *Lateral Movement* inference, while maintaining inter-packet timing enables *Execution* inference. Rather than exporting all possible metrics, operators can use the audit framework to identify the minimal telemetry schema that supports specific threat hypotheses.

For instance, to preserve *Lateral Movement* inference in a P4 switch, our audit dictates exporting the *sourceIP-destinationIP* matrix per time bin, not just aggregate counters.

Similarly, for IPFIX template design, operators must include *flowStartMilliseconds* and *flowEndMilliseconds* fields (rather than only aggregate byte/packet counts) to enable inter-packet timing reconstruction for *Execution* inference. This is particularly relevant for IPFIX extensions and custom telemetry formats, where operators must balance export bandwidth against forensic utility.

6.7. Feasibility Analysis: Integration into SDN Controllers and P4 Pipelines

To address practical implementation concerns, we provide a detailed feasibility analysis for integrating the audit framework into real-world SDN controllers and P4-programmable switch pipelines. This analysis outlines architectural considerations, implementation challenges, and concrete integration strategies.

6.7.1. SDN Controller Integration Architecture

The audit framework can be integrated into SDN controllers (e.g., OpenDaylight, ONOS, Ryu) as a *telemetry policy engine* that evaluates and enforces forensic coverage requirements. The proposed architecture positions the audit framework as a middleware component between the controller's northbound API and southbound telemetry collection.

Architecture Components:

- **Policy Definition Module:** Allows operators to specify required ATT&CK tactic coverage (e.g., "preserve support for Lateral Movement and Execution tactics") and acceptable abstraction levels.
- **Real-time Audit Engine:** Evaluates current telemetry exports against policy requirements, computing artifact computability and inference coverage metrics.
- **Adaptive Telemetry Manager:** Dynamically adjusts telemetry granularity (packet sampling rates, flow timeout settings, aggregation bin sizes) based on audit results and resource constraints.
- **Forensic Coverage Monitor:** Continuously tracks which threat hypotheses remain supportable given the current telemetry configuration.

Implementation Challenges:

Challenge 1: Real-time Audit Computation Overhead. The audit framework must evaluate artifact computability across multiple evidence layers in real-time. For a network with thousands of switches, this could introduce significant computational overhead. **Solution:** Implement incremental audit updates that recompute metrics only when the telemetry configuration changes, rather than continuously re-evaluating all artifacts. Cache artifact computability results per switch/port configuration, invalidating only when relevant parameters (timeout settings, bin sizes) are modified.

Challenge 2: Controller-Switch Communication Latency. SDN controllers communicate with switches via OpenFlow or P4Runtime, introducing latency between policy decisions and telemetry reconfiguration. This delay could create windows where forensic coverage requirements are temporarily unmet. **Solution:** Implement predictive policy enforcement that pre-configures telemetry settings based on anticipated traffic patterns, and use asynchronous audit validation that tolerates brief coverage gaps during reconfiguration.

Challenge 3: Heterogeneous Switch Capabilities. Different switch models support varying telemetry features (e.g., some support packet mirroring, others only flow export). The audit framework must adapt to these constraints. **Solution:** Maintain a capability matrix per switch type, mapping available telemetry features to artifact computability. The audit engine queries this matrix to determine feasible coverage levels per device.

Challenge 4: Policy Conflict Resolution. Operators may specify conflicting requirements (e.g., "preserve Execution inference" while "minimize export bandwidth"). The framework must resolve these trade-offs. **Solution:** Implement a priority-based policy resolution

system that ranks tactics by operational importance (e.g., Critical/High/Medium), and uses the audit framework to find minimal telemetry configurations that satisfy critical requirements first.

6.7.2. P4 Pipeline Integration Architecture

For P4-programmable switches, the audit framework informs *the design of the compile-time telemetry schema and the runtime selective export policies*. Unlike SDN controllers, which operate in the control plane, P4 integration requires embedding audit logic directly into the data-plane pipeline.

Architecture Components:

- **Schema Generator:** Uses audit framework output to generate P4 telemetry table definitions that preserve required artifacts (e.g., registers for per-source destination fan-out, timestamps for inter-packet intervals).
- **Selective Export Logic:** Implements conditional telemetry export based on artifact requirements—only exporting detailed metrics when they enable critical tactic support.
- **Resource-Aware Binning:** Dynamically adjusts temporal aggregation bin sizes based on available switch memory and required artifact preservation.

Implementation Challenges:

Challenge 1: P4 Memory Constraints. P4 switches have limited memory for stateful data structures (registers, meters). Storing per-source destination fan-out matrices or fine-grained timing data for all flows may exceed available resources. **Solution:** Implement approximate data structures (e.g., Count-Min sketches for fan-out estimation, sampled timing windows) that preserve artifact computability within memory bounds. The audit framework can be extended to evaluate artifact computability under approximation, trading exactness for feasibility.

Challenge 2: Pipeline Stage Limitations. P4 pipelines have fixed processing stages, limiting where telemetry logic can be inserted. Complex artifact computations (e.g., temporal periodicity detection) may require multiple pipeline passes. **Solution:** Decompose artifact computation into pipeline-compatible primitives. For example, periodicity detection can be implemented using P4 registers that track flow counts per time bin, with periodicity analysis performed by a control plane agent that processes exported register snapshots.

Challenge 3: Export Bandwidth Constraints. High-speed switches generate massive telemetry volumes. Exporting detailed artifacts (e.g., per-packet timestamps) could saturate control plane links. **Solution:** Use the audit framework to identify *minimal sufficient* artifact sets. For example, if both “inter-arrival burstiness” and “inter-packet timing” support Execution, but only one is required, export only the less bandwidth-intensive artifact. Implement intelligent sampling that exports detailed metrics only for flows matching suspicious patterns (e.g., high fan-out, SYN-dominant).

Challenge 4: Compile-Time vs. Runtime Flexibility. P4 programs are compiled and loaded onto switches, limiting runtime reconfiguration. However, telemetry requirements may change based on threat intelligence. **Solution:** Design P4 pipelines with *parameterized* telemetry tables (e.g., configurable bin sizes, selectable export fields) that can be modified via P4Runtime without recompilation. The audit framework validates that parameter changes maintain required coverage.

6.7.3. Concrete Integration Example: P4 Switch with Hybrid Monitoring

To illustrate practical integration, we outline a concrete P4 implementation for a switch that must preserve support for *Lateral Movement* and *Execution* tactics while operating under memory constraints.

Requirements:

- Preserve *Lateral Movement* inference (requires destination fan-out artifact).
- Preserve *Execution* inference (requires inter-packet timing artifact).
- Memory budget: 64 KB for telemetry state.
- Export bandwidth: <1% of data plane throughput.

P4 Implementation Strategy:

1. **Destination Fan-out Preservation:** Use a P4 register array indexed by source IP hash, storing a compact representation of destination sets (e.g., Bloom filter or Count-Min sketch). Export register snapshots every 10 s, enabling per-source fan-out computation at the control plane.
2. **Inter-packet Timing Preservation:** For flows matching suspicious patterns (SYN-dominant, short-lived), enable packet-level timestamp export. Use P4 m to identify candidate flows, then mirror selected packets to a telemetry collector with full timing information.
3. **Resource Management:** Implement LRU eviction for fan-out registers when memory is exhausted, prioritizing high-activity sources. Use the audit framework to validate that eviction policies do not eliminate support for critical tactics.

Validation: The audit framework evaluates this configuration and confirms that (i) destination fan-out remains computable from exported register snapshots (L1-level support), and (ii) inter-packet timing is available for selected flows (L0-level support for Execution). The framework reports that *Lateral Movement* and *Execution* remain supportable under this hybrid configuration, while *Persistence* (requiring entity-linked periodicity) becomes non-supportable due to register eviction, which removes long-term state.

6.7.4. Scalability and Computational Overhead Analysis

To address concerns about scalability and computational overhead for large-scale, high-speed backbone traffic over extended periods, we provide a *complexity- and pipeline-oriented* analysis. We do *not* claim a universal benchmark (e.g., a fixed number of minutes per 15 min window), because runtime depends strongly on traffic volume, capture settings, hardware, and implementation details.

Audit Framework Overhead: The audit framework's core operation—checking artifact computability against a telemetry schema—is a lightweight, rule-based process that evaluates structural properties of evidence representations. The computationally intensive work (PCAP to L1/L2 conversion, flow record generation, time aggregation) is performed by standard tools (e.g., YAF [41]) as part of the normal telemetry pipeline deployment. The audit itself is a *meta-analysis* of the output schema, not a full traffic re-analysis. Once artifact computability is determined for a given telemetry configuration (L0/L1/L2), the audit results remain valid regardless of traffic volume or the length of the time period, making the framework highly scalable for design-time evaluation.

Computational Complexity (High Level):

The audit framework's computational overhead is dominated by artifact computation and depends on the evidence product:

- **L0 (Packet-level):** Processing scales linearly with the number of packets whose headers/timestamps are parsed and used for artifact computation ($O(n)$ with respect to packet count). This layer is the most computationally demanding and is typically infeasible to run continuously at line rate on high-speed backbones without dedicated capture infrastructure, sampling, or selective capture.
- **L1 (Flow-level):** Processing scales with the number of exported flow records ($O(f)$ with respect to flow count). In operational settings, flow generation is performed by ex-

porters/meters (e.g., IPFIX exporters), so the audit can be applied to already-exported flow logs, shifting the primary cost from packet processing to flow-log analytics.

- **L2 (Time-aggregated):** Processing scales with the number of time bins and exported aggregate counters ($O(t)$ with respect to bins/records). Because aggregation substantially reduces volume, L2 audits are typically the least computationally intensive.

Scaling to Extended Time Periods:

For extended periods (hours, days, weeks), the framework can be applied using several strategies:

Strategy 1: Incremental Processing. The audit framework evaluates artifact *computability*, which is a structural property of the representation, not traffic volume. Once artifact computability is established for a telemetry schema (L0/L1/L2), it remains invariant regardless of time period length. Therefore, the audit can be performed once per telemetry configuration, not continuously over time. For extended periods, operators can do the following:

- Run the audit on representative time windows (e.g., peak hours, off-peak hours).
- Cache artifact computability results per telemetry configuration.
- Re-run the audit only when telemetry schema changes (e.g., IPFIX template modifications, aggregation bin size adjustments).

Strategy 2: Sampling-Based Evaluation. For very large datasets, the audit can be applied to sampled subsets. Since artifact computability is representation-driven (not traffic-dependent), sampling does not affect the structural results. Operators can do the following:

- Sample representative time windows (e.g., 1-h samples per day).
- Apply the audit to sampled data to validate artifact computability.
- Extrapolate results to the full time period, since computability is invariant to traffic volume.

Strategy 3: Distributed Processing. For multi-terabyte archives, artifact computation can be parallelized:

- Partition data by time windows or network segments.
- Compute artifacts independently per partition.
- Aggregate results (artifact computability is identical across partitions for the same representation).

Memory and Storage Requirements (Qualitative):

Memory requirements depend on whether artifacts are computed in a streaming fashion or require maintaining per-entity state (e.g., per-source adjacency surrogates for fan-out, per-entity periodicity features). In practice, the audit can be implemented as a streaming pipeline over PCAP/flow logs, with bounded memory determined by the chosen artifact set and any required state (e.g., hash tables, sketches, sliding windows). L2 representations generally have the smallest storage and memory footprint because they are already aggregated.

High-Speed Traffic Considerations:

For high-speed backbone links, the primary bottleneck is typically data ingestion and storage at L0 (packet capture), not the audit logic itself:

- **Packet-level (L0):** Continuous full-rate packet capture is often infeasible at backbone scale due to storage and processing constraints; in such settings, operators rely on sampling, selective capture, or derived products. Our audit can be applied to sampled or selectively captured packet traces to validate which packet-level artifacts are computable under the deployed capture policy.
- **Flow-level (L1):** Flow exporters (e.g., IPFIX) handle high-speed traffic natively. The audit framework processes exported flow records that are already aggregated by the exporter, thereby avoiding packet-level bottlenecks.

- **Time-aggregated (L2):** Pre-aggregated statistics from high-speed links are typically manageable volumes, enabling efficient audit processing.

Practical Recommendations:

For large-scale, extended-period deployments:

1. **Design-time evaluation:** Run the audit once per telemetry configuration during design/planning phases, not continuously during operation.
2. **Configuration change triggers:** Re-run the audit only when telemetry schemas change (IPFIX template updates, aggregation parameter modifications).
3. **Representative sampling:** For validation, apply the audit to representative time windows rather than processing entire archives.
4. **Incremental processing:** Use streaming algorithms for artifact computation to enable constant-memory processing of extended periods.
5. **Caching:** Cache artifact computability results per telemetry configuration, invalidating only when relevant parameters change.

Limitations:

The current implementation processes 15 min windows sequentially. For very large archives (months/years), full processing would require the following:

- Significant computational resources (days/weeks of CPU time for packet-level processing).
- Large storage capacity for intermediate results.
- However, this is typically unnecessary since artifact computability is invariant to time period length once the telemetry schema is fixed.

Future work should develop optimized implementations using distributed processing, approximate algorithms for very large datasets, and hardware acceleration for packet-level processing on high-speed links.

6.7.5. Implementation Limitations and Future Work

Several limitations must be acknowledged for realistic deployment:

Platform-Specific Constraints: The feasibility analysis assumes standard P4 and SDN capabilities, but real deployments may face vendor-specific limitations (e.g., restricted register sizes and fixed export formats). Future work should develop platform-specific audit adapters that map framework outputs to vendor capabilities.

Dynamic Traffic Adaptation: The current framework evaluates static telemetry configurations. In practice, traffic patterns change, requiring dynamic reconfiguration. Future work should extend the framework to support online adaptation, continuously monitoring coverage metrics, and triggering reconfiguration when thresholds are breached.

Multi-Switch Coordination: Large networks require coordinated telemetry policies across multiple switches. The framework currently evaluates single-device configurations. Future work should develop distributed audit protocols that ensure network-wide coverage requirements are met while respecting per-device resource constraints.

Scalability for Extended Periods: While the framework's computational complexity is manageable for design-time evaluation, processing multi-terabyte archives would require significant resources. However, since artifact computability is invariant to the length of the time period (once the telemetry schema is fixed), full archive processing is typically unnecessary. Future work should develop optimized implementations for very large-scale deployments.

This feasibility analysis demonstrates that while integration is non-trivial, the audit framework provides actionable guidance for real-world deployment, with concrete solutions to identified implementation challenges.

6.8. Handling Partial and Borderline Support Cases

While our framework evaluates artifact *computability* as a binary property (supportable vs. non-supportable), practitioners must often reason about cases where artifacts gradually deteriorate rather than completely vanish. This subsection provides structured guidance for interpreting and acting upon partial or borderline support cases.

6.8.1. Degradation Categories

We classify artifact degradation into three categories:

Complete Loss: The artifact cannot be computed from the representation due to fundamental information loss. For example, inter-packet timing patterns are completely lost at L1 because flow records do not contain per-packet timestamps. In such cases, the artifact provides no support for tactics that require it (e.g., Execution inference).

Precision Degradation: The artifact remains technically computable but with reduced precision or reliability. For example, entity-linked periodicity at L1 may be affected by flow timeout effects that split periodic patterns across multiple flow records. Practitioners must evaluate whether the degraded precision is sufficient for their forensic objectives.

Scope Reduction: The artifact remains computable but for a reduced set of entities or time windows. For example, destination fan-out at L1 may lose precision if flow timeouts cause connection splitting, reducing the reliability of Lateral Movement inference. Practitioners must assess whether the reduced scope enables actionable forensic conclusions.

6.8.2. Decision Framework for Practitioners

We provide a structured decision framework that helps practitioners evaluate whether partial support is sufficient:

High-Stakes Scenarios: When forensic defensibility is critical (e.g., legal proceedings, compliance audits), practitioners should treat precision-degraded artifacts as non-supportable to maintain conservative reasoning. For example, if entity-linked periodicity at L1 is affected by flow timeout splitting, practitioners should not rely on it for Persistence inference in legal contexts.

Operational Monitoring: When the objective is operational threat detection rather than forensic reconstruction, precision-degraded artifacts may be acceptable if they enable actionable alerts. For example, destination fan-out at L1 may be sufficient for operational Lateral Movement detection even if flow timeouts reduce precision, as long as alerts trigger further investigation.

Hybrid Configurations: When partial support is insufficient, practitioners can design hybrid monitoring configurations that preserve critical artifacts at higher-fidelity layers. For example, selective packet capture (L0) for high-value targets while using flow records (L1) for general monitoring enables both scalable operation and forensic coverage for critical assets.

6.8.3. Degradation Indicators

We introduce qualitative degradation indicators that practitioners can use to assess artifact reliability:

Flow Timeout Effects: For artifacts that depend on connection continuity (e.g., entity-linked periodicity), practitioners should evaluate whether flow timeout settings cause significant splitting that degrades artifact reliability. Extended flow timeouts (e.g., 300 s) may preserve periodic patterns better than short timeouts (e.g., 15 s), but at the cost of increased state overhead.

Aggregation Window Effects: For time-aggregated artifacts, practitioners should assess whether aggregation windows are sufficiently fine-grained to preserve temporal

patterns of interest. For example, 1-s bins may preserve Execution-related timing patterns better than 60-s bins, but at the cost of increased storage.

Sampling Effects: When sampling is used (e.g., packet sampling at L0), practitioners should evaluate whether the sampling rate provides sufficient coverage for artifact computation. For example, 1:100 packet sampling may preserve rate-based artifacts but may miss low-volume timing patterns required for Execution inference.

6.8.4. Practical Examples

Example 1: Lateral Movement Inference with Flow Timeout Effects

Consider a scenario where practitioners must evaluate whether L1 flow records provide sufficient support for Lateral Movement inference when destination fan-out may be affected by flow timeout splitting. The decision framework suggests the following:

- **High-Stakes:** If forensic defensibility is critical, treat destination fan-out at L1 as non-supportable due to potential timeout effects, and require L0 packet capture or extended flow timeouts.
- **Operational:** If the objective is operational detection, destination fan-out at L1 may be sufficient if flow timeouts are configured appropriately (e.g., 60-s active timeout) and alerts trigger further investigation.
- **Hybrid:** Use L1 for general monitoring with selective L0 packet capture for suspicious flows, enabling both scalable operation and forensic coverage.

Example 2: Persistence Detection with Entity-Linked Periodicity

Consider a scenario where practitioners must determine whether L2 time-aggregated statistics enable Persistence detection when entity-linked periodicity is lost but aggregate periodicity remains. The decision framework suggests the following:

- **High-Stakes:** Entity-linked periodicity is required for Persistence inference; aggregate periodicity alone is insufficient. Practitioners should not rely on L2 for Persistence inference in legal contexts.
- **Operational:** Aggregate periodicity at L2 may enable operational detection of periodic traffic patterns, but cannot attribute periodicity to specific entities. Practitioners should use L2 for initial alerts and require L1 or L0 for entity attribution.
- **Hybrid:** Use L2 for general monitoring with selective L1 flow records for entities exhibiting periodic patterns, enabling both scalable operation and entity-level attribution.

This structured guidance enables practitioners to make informed decisions about partial support cases while maintaining the framework's focus on structural computability as the primary evaluation criterion.

6.9. Practical Ramifications and Telemetry Design Trade-Offs

While our results are well-interpreted at a high level, practitioners require detailed guidance on concrete trade-offs when designing telemetry systems. This subsection provides actionable analysis of practical ramifications and design decision-making.

6.9.1. Concrete Design Trade-Off Analysis

Trade-off 1: Storage vs. Forensic Coverage

Scenario: An ISP must choose between L0 packet capture (high storage cost, full forensic coverage) vs. L1 flow records (low storage cost, reduced forensic coverage).

Trade-off Analysis:

- **L0:** Provides support for 9 ATT&CK tactics but requires substantially higher storage compared to L1 (packet headers vs. flow summaries). The storage ratio depends on

traffic characteristics, but packet-level capture typically requires orders of magnitude more storage than flow records.

- **L1:** Provides support for 8 ATT&CK tactics but loses Execution inference capability (requires packet-level timing).
- **L2:** Provides support for 7 ATT&CK tactics but enables flow-based defensive techniques not available at L0 (e.g., Flow-based Monitoring).

Practical Recommendation: For resource-constrained switches, prioritize L1 with extended flow timeouts (e.g., 300 s) to preserve entity-linked artifacts while avoiding per-packet overhead. Use approximate data structures (e.g., Count-Min sketches) for high-cardinality artifacts (e.g., destination fan-out) to manage memory constraints.

Trade-off 3: Scalability vs. Defensive Coverage

Scenario: An enterprise network must choose a monitoring architecture for 10 Gbps links with varying security requirements.

Trade-off Analysis:

- **L0:** Enables full defensive coverage (9 tactics) but may not scale to high-speed links without sampling. Continuous full-rate packet capture at 10 Gbps requires specialized hardware and significant storage.
- **L1:** Scales to high-speed links but loses Execution inference and some behavioral monitoring capabilities. Flow exporters handle 10 Gbps natively with standard hardware.
- **L2:** Scales best but loses entity-linked periodicity and destination fan-out (affecting Persistence and Lateral Movement). Pre-aggregated statistics are manageable even at 100+ Gbps.

Practical Recommendation: Use L1 as a baseline with selective L0 sampling for suspicious flows, enabling scalable operation while preserving critical forensic capabilities. For example, use L1 flow records for all traffic and trigger L0 packet capture (1:100 sampling or selective mirroring) for flows matching threat intelligence indicators or anomaly detection alerts.

6.9.2. Non-Monotonic Behavior Implications

The non-monotonic transformation of defensive coverage (D3FEND applicability increases at L1 before decreasing at L2) has important practical implications:

Why D3FEND Applicability Increases at L1: Flow-based monitoring techniques become enabled at L1 (requiring flow records), partially compensating for the loss of packet-level behavioral monitoring. For example, Flow-based Monitoring enables defensive techniques (e.g., flow-based rate limiting, connection throttling) that are not applicable at L0, where only packet-level behavioral monitoring is available.

Practical Implication: Practitioners should not assume that more abstract telemetry always reduces defensive capabilities. Instead, they should evaluate which defensive techniques are enabled/disabled at each layer. For example, L1 may be preferable to L0 for certain defensive objectives (e.g., flow-based DDoS mitigation) even though it loses Execution inference capability.

Design Guidance: When designing hybrid monitoring systems, practitioners can strategically combine L0, L1, and L2 to maximize defensive coverage by enabling complementary defensive techniques. For example, use L0 for behavioral monitoring (Execution, fine-grained anomaly detection) and L1 for flow-based defensive techniques (rate limiting, connection throttling), enabling both capabilities simultaneously.

6.9.3. Concrete Telemetry Configuration Examples

Example 1: ISP Backbone Monitoring

Constraints: 100 Gbps links, 30-day retention, privacy regulations limit payload access.

Audit Application: Evaluate L1 vs. L2 for different network segments.

Decision: Use L1 for general monitoring (preserves 8 tactics and enables flow-based defense), and L2 for long-term archival (reduces storage and preserves 7 tactics). For example, retain L1 flow records for 7 days (for operational monitoring) and L2 time-aggregated statistics for 30 days (for long-term analysis).

Risk Assessment: Loss of Execution inference at both L1 and L2 is acceptable for the ISP context, as endpoint monitoring (outside ISP scope) handles Execution detection. The ISP's primary forensic objectives (Lateral Movement, Command and Control, Exfiltration) remain supportable at L1.

Example 2: Enterprise Network Security Operations Center (SOC)

Constraints: 10 Gbps links, real-time threat detection required, forensic reconstruction needed for incident response.

Audit Application: Evaluate hybrid L0 + L1 configuration.

Decision: Use L1 for real-time monitoring (scalable, enables 8 tactics) with selective L0 packet capture for suspicious flows (enables Execution inference, preserves full forensic coverage). For example, use L1 flow records for all traffic and trigger L0 packet capture (selective mirroring) for flows matching threat intelligence indicators or anomaly detection alerts.

Risk Assessment: Hybrid configuration balances scalability with forensic coverage, enabling both operational detection and post-incident reconstruction. The SOC can investigate Execution-related incidents using L0 packet captures while maintaining scalable L1 monitoring for general traffic.

Example 3: Cloud Provider Network Monitoring

Constraints: Multi-tenant environment, varying security requirements, programmable data plane (P4) available.

Audit Application: Evaluate P4 telemetry export schema design.

Decision: Design a parameterized P4 pipeline that can export L0, L1, or L2 based on tenant security requirements, enabling dynamic telemetry configuration. For example, high-security tenants can request L0 packet capture (full forensic coverage), standard tenants receive L1 flow records (balanced coverage and cost), and low-security tenants receive L2 time-aggregated statistics (cost-optimized).

Risk Assessment: Parameterized design enables tenants to choose the telemetry abstraction level based on their security needs and cost constraints. The cloud provider can offer tiered monitoring services (Premium: L0, Standard: L1, Basic: L2) with corresponding pricing and forensic coverage.

6.9.4. Cost-Benefit Framework

We provide a structured framework for practitioners to evaluate telemetry design decisions:

Cost Dimensions:

- **Storage:** Data retention costs (L0: high, L1: moderate, L2: low).
- **Processing Overhead:** Computational requirements (L0: high, L1: moderate, L2: low).
- **Export Bandwidth:** Network overhead for telemetry export (L0: high, L1: moderate, L2: low).
- **Implementation Complexity:** Development and maintenance effort (L0: high, L1: moderate, L2: low).

Benefit Dimensions:

- **Forensic Coverage:** Number of supportable ATT&CK tactics (L0: 9, L1: 8, L2: 7).
- **Defensive Applicability:** Number of enabled D3FEND techniques (L0: 7, L1: 8, L2: 7).

- **Operational Capabilities:** Real-time detection, incident response, and forensic reconstruction.

Decision Matrix: Practitioners can use this framework to evaluate trade-offs between cost and benefit dimensions for different telemetry configurations, enabling data-driven design decisions informed by organizational priorities and constraints.

6.9.5. Integration with Existing Systems

IPFIX Exporter Configuration: Practitioners can configure flow timeout settings to preserve entity-linked artifacts while managing state overhead. For example, extended active timeouts (300 s) preserve periodic patterns better than short timeouts (15 s), but at the cost of increased flow state memory. The audit framework can evaluate which timeout settings preserve required artifacts for specific forensic objectives.

Time-Aggregation Policy: Practitioners can choose aggregation windows that balance storage reduction with temporal pattern preservation. For example, 1-s bins preserve Execution-related timing patterns better than 60-s bins, but at the cost of increased storage. The audit framework can evaluate which aggregation windows preserve required artifacts for specific forensic objectives.

Selective Capture Strategies: Practitioners can design sampling or selective capture policies that preserve critical artifacts while managing resource constraints. For example, 1:100 packet sampling may preserve rate-based artifacts but may miss low-volume timing patterns. The audit framework can evaluate which sampling strategies preserve required artifacts for specific forensic objectives.

This expanded discussion provides concrete, actionable guidance that practitioners can directly apply to their telemetry design decisions, moving beyond high-level interpretation to practical implementation strategies.

6.10. Limitations and Future Work

This study intentionally focused on backbone-level passive observation without payload visibility, endpoint context, or ground truth. While this reflects common operational constraints, several limitations warrant future investigation.

Artifact Catalog Scope: The 13 artifacts examined in this study are *representative* rather than exhaustive, designed to demonstrate the framework's methodology and enable evaluation of all network-observable ATT&CK tactics. The framework is extensible: The mapping method is general and can accommodate additional artifacts (e.g., DNS query patterns, TLS handshake features, packet size distributions) as they are identified in future work. The current instantiation using 13 artifacts provides sufficient coverage to demonstrate selective and structural inference loss patterns, but practitioners may extend the catalog based on domain-specific requirements.

Other Limitations: First, the artifact catalog is currently static; future work could develop adaptive artifact definitions that adjust based on traffic characteristics or threat intelligence. Second, the analysis assumes deterministic artifact computation; in practice, measurement noise, sampling, and flow timeout variations introduce uncertainty that should be quantified. Third, the framework evaluates artifact *computability* but not evidential *strength*; future work could develop probabilistic models that quantify confidence in tactic support given noisy or partial observations.

The detailed feasibility analysis provided in Section 6 (Feasibility Analysis: Integration into SDN Controllers and P4 Pipelines) addresses implementation challenges and concrete integration strategies for real-world deployment.

7. Conclusions

We developed a design-time audit tool that identifies how progressively abstracted network evidence alters the set of threat hypotheses and defensive actions that can be logically supported. Using backbone traffic from MAWI DITL [8] (2025) and a literature-grounded inference framework, the audit procedure revealed that evidence abstraction results in *selective and structural inference losses*. Specifically, *Execution* becomes non-supportable at L1 (due to the loss of packet-level timing artifacts required for execution-related network manifestations), while Lateral Movement and Persistence become non-supportable at L2 (due to the loss of entity-linked structural artifacts, such as destination fan-out and host-specific periodicity). In contrast, tactics such as *Reconnaissance*, *Discovery*, and *Command and Control* remain supportable across all representations, albeit with reduced evidential strength.

The audit further revealed that defensive reasoning does not degrade monotonically with abstraction. Instead, abstraction *transforms* the toolkit of defensible support. Flow export enables flow-based monitoring capabilities, while coarse aggregation disables entity-aware anomaly detection and behavioral analysis. This shift reflects a transition from fine-grained behavioral reasoning to rate- and flow-based control strategies, rather than a uniform loss of defensive capability.

The audit framework provides organizations with a method to evaluate their telemetry architectures. Organizations should align their incident response playbooks, forensic reporting practices, and defensive expectations with the abstraction level of their network telemetry. Claims about adversary behavior that require packet timing or entity-linked structure become non-supportable when monitoring systems export only aggregated flow statistics. Monitoring architectures should document not only what data is collected, but also which forensic inference classes remain supportable under the audit framework.

This study intentionally focused on backbone-level passive observation without payload visibility, endpoint context, or ground truth. The evaluation is limited to a single backbone dataset (MAWI DITL), thereby limiting the generalizability of the empirical findings. However, the framework's structural analysis (artifact computability) is representation-driven and should generalize across environments, as artifact computability is determined by evidence representation structure rather than traffic characteristics. Future work could extend the analysis to hybrid telemetry environments, enterprise and data center networks, and assess evidential strength under adversarial traffic injection.

In summary, this work provides an audit methodology that reframes evidence reduction not as a loss of data volume, but as a loss of inferential options. While prior work optimizes for detection accuracy (measuring how well attacks are identified) or telemetry efficiency (optimizing export performance), this work provides a prerequisite analysis for forensic defensibility, enabling informed trade-offs between scalability and inference capability. The framework enables organizations to evaluate which forensic claims remain supportable under their chosen telemetry architecture, which is essential for designing monitoring systems, qualifying forensic conclusions, and maintaining analytical rigor in modern network security operations.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/network6010009/s1>. Supplementary File S1: *decision_rules.json* (Machine-readable structured decision rules encoding explicit if-then criteria for artifact-to-tactic supportability, enabling independent validation of mapping decisions). Supplementary File S2: *audit_framework_code.zip* (Artifact computation scripts and audit framework implementation, including configuration files, dependency specifications, and step-by-step run instructions for reproducing the analysis using publicly available MAWI data). Supplementary File S3: *artifact_catalog.json* (Machine-readable artifact catalog documenting all 13 network-forensic

artifacts, their computability across evidence layers, associated ATT&CK tactic-level relationships, and full literature citations).

Author Contributions: Conceptualization, M.V.; methodology, M.V. and S.J.; software, M.V.; validation, S.J. and H.N.; formal analysis, M.V.; investigation, M.V.; resources, M.V.; data curation, M.V.; writing—original draft preparation, M.V.; writing—review and editing, S.J. and H.N.; visualization, M.V.; supervision, S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The MAWI Day-In-The-Life (DITL) dataset used in this study is publicly available from the MAWI Working Group Traffic Archive [8] at <https://mawi.wide.ad.jp/mawi/>, (accessed on 5 December 2025). The specific 15 min PCAP files analyzed in this study (9 April 2025; 00:00, 06:00, 12:00) can be downloaded from: <http://mawi.nezu.wide.ad.jp/mawi/ditl/ditl2025/202504090000.pcap.gz> (accessed on 5 December 2025), <http://mawi.nezu.wide.ad.jp/mawi/ditl/ditl2025/202504090600.pcap.gz> (accessed on 5 December 2025), and <http://mawi.nezu.wide.ad.jp/mawi/ditl/ditl2025/202504091200.pcap.gz> (accessed on 5 December 2025). The artifact computation scripts, audit framework code, and machine-readable decision rules are provided as Supplementary Materials (Files S1–S3).

Acknowledgments: During the preparation of this manuscript, the authors used AI-assisted tools for language proficiency and text editing purposes only. All ideas, technical content, analysis, methodology, and research contributions are the original work of the authors. The authors have reviewed and edited all AI-assisted output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
D3FEND	Detection, Denial, and Disruption Framework Emulating Network Defense
IPFIX	IP Flow Information Export
ISP	Internet Service Provider
PCAP	Packet Capture
SOC	Security Operations Center
IR	Incident Response
SDN	Software-Defined Networking
DDoS	Distributed Denial of Service
NIDS	Network Intrusion Detection System
TLS	Transport Layer Security
DITL	Day-In-The-Life
YAF	Yet Another Flowmeter

Appendix A. Artifact Catalog

The complete artifact catalog documents all 13 network-forensic artifacts, their computability across evidence layers (L0, L1, L2), associated ATT&CK tactic-level support relationships, and enabling D3FEND defensive technique categories. Each artifact-to-tactic association is grounded in published literature, with source citations provided. This catalog serves as the reference for all survivability and applicability analyses reported in the main text.

Table A1 provides the complete mapping. Artifacts are defined operationally (exact computable functions) to ensure reproducibility. The “Available In” column indicates which

evidence layers support artifact computation, while “Supports Tactics” lists ATT&CK tactics for which published sources describe consistent network-visible behavioral relationships. “Enables D3FEND” identifies defensive technique categories that become actionable when the artifact is observable.

Table A1. Complete artifact catalog with literature-grounded associations. Each artifact is operationally defined and mapped to ATT&CK tactics and D3FEND categories based on published sources.

Artifact	Available In	Supports Tactics	Enables D3FEND
Inter-arrival Time Burstiness	L0	Execution, Command and Control	Behavioral Monitoring
Inter-packet Timing Patterns	L0	Execution, Command and Control	Behavioral Monitoring
SYN-Dominant Connection Bursts	L0, L1, L2	Reconnaissance, Initial Access, Discovery	Connection Throttling, Traffic Filtering
Packet Rate Spikes	L0, L1, L2	Execution, Impact, Command and Control	Rate Limiting, Traffic Filtering
Byte Rate Spikes	L0, L1, L2	Exfiltration, Impact	Rate Limiting, Traffic Filtering
Destination Fan-out	L0, L1	Lateral Movement, Discovery	Network Traffic Analysis
Temporal Periodicity (entity-linked)	L0, L1	Persistence, Command and Control	Anomaly Detection, Behavioral Monitoring
Aggregate Temporal Periodicity	L0, L1, L2	Command and Control	Behavioral Monitoring
Flow Duration Distribution	L1, L2	Execution, Discovery	Flow-based Monitoring
Directional Traffic Asymmetry	L1, L2	Exfiltration, Lateral Movement	Network Traffic Analysis
Short-Lived Flow Patterns	L1, L2	Reconnaissance, Discovery	Flow-based Monitoring
Protocol Distribution Imbalance	L0, L1, L2	Execution, Discovery	Protocol Analysis
ICMP Protocol Share	L0, L1, L2	Discovery, Reconnaissance	Protocol Analysis

Note: The complete catalog with detailed artifact definitions, computation procedures, and full citation lists for each association is available in the Supplementary Materials as a machine-readable JSON file (`artifact_catalog.json`) and as a detailed spreadsheet.

For each artifact-tactic pair, the structured decision rules (Table 5 in Section 4) provide explicit if-then criteria for determining supportability based on evidence property requirements and literature-grounded associations. Complete decision rules for all 13 artifacts with full citation lists, including detailed application examples and uncertainty handling procedures, are available in machine-readable format (`decision_rules.json`) in the Supplementary Materials. This structured approach enables independent validation of mapping decisions and enhances reproducibility.

Appendix B. Stability Across Time Windows

Section 4 reports that the audit procedure produces consistent collapse patterns across all three time windows (00:00, 06:00, 12:00), with identical metrics at each layer. This appendix provides extended visualizations demonstrating this invariance.

Figures A1 and A2 show the collapse metrics for the 06:00 and 12:00 windows, respectively. These figures replicate the structure of Figure 5 (main text, 00:00 window) and confirm that the non-monotonic transformation pattern—artifact count increase at L1, monotonic inference coverage decrease, and D3FEND applicability transformation—is consistent across diverse traffic conditions.

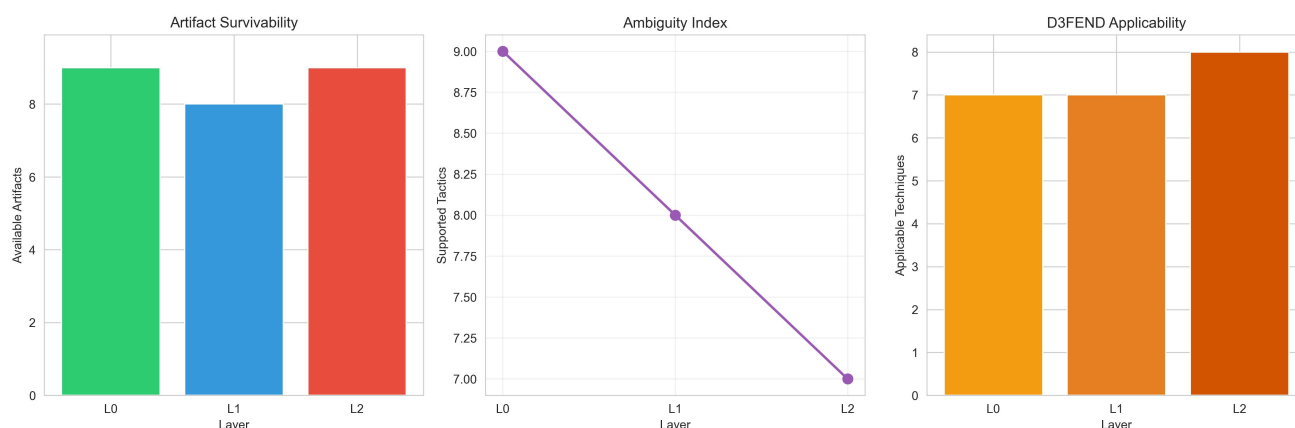


Figure A1. Collapse metrics for the 06:00 time window, replicating the analysis shown in Figure 5 (main text). The identical pattern (10/11/9 artifacts, 9/8/7 tactics, 7/8/7 D3FEND categories) confirms that evidence-loss effects are structural properties of monitoring abstractions rather than artifacts of specific traffic characteristics.

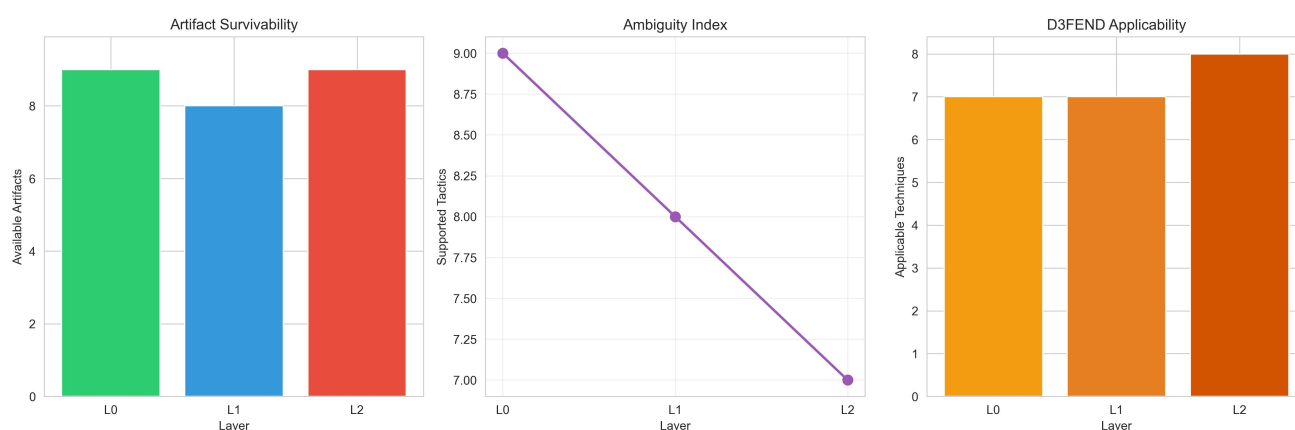


Figure A2. Collapse metrics for the 12:00 time window, again showing identical patterns to the 00:00 and 06:00 windows. This consistency across night (00:00), morning (06:00), and noon (12:00) periods indicates that the audit framework produces stable results across the evaluated traffic conditions.

Appendix C. Extended Illustrative Outputs

This section provides additional visualizations that instantiate the artifact survivability, inference coverage (ambiguity index), and D3FEND applicability metrics across all three time windows. These figures extend the tabular results reported in the main text (Tables 6–8) by showing the layer-by-layer progression for each metric.

Appendix C.1. Artifact Survivability

Figures A3 and A4 show artifact survivability for the 06:00 and 12:00 windows, complementing the pattern established in the main text. The consistent pattern across windows—10 artifacts at L0, 11 at L1 (reflecting emergence of flow-specific properties), and 9 at L2—validates that artifact computability is representation-driven rather than traffic-dependent.

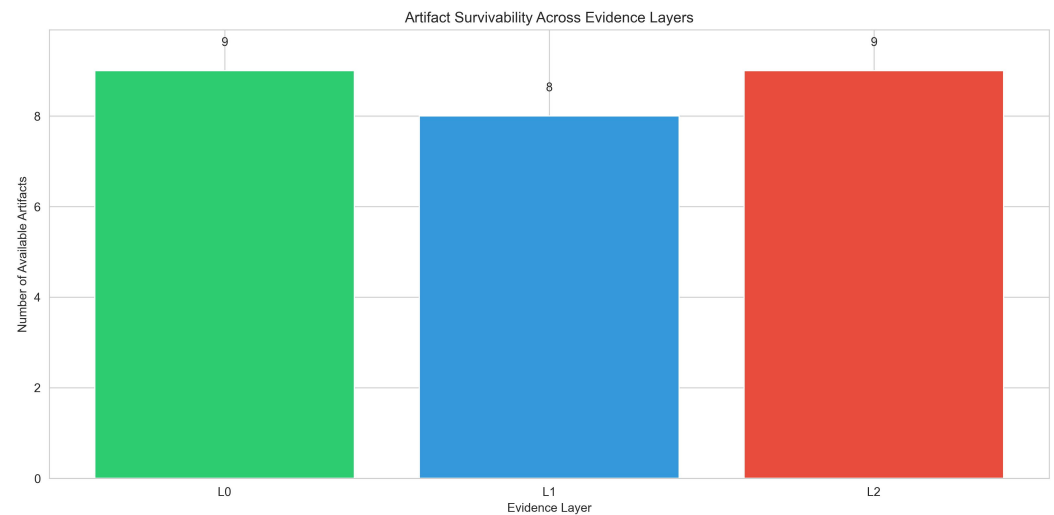


Figure A3. Artifact survivability across evidence layers for the 06:00 time window. The pattern matches the 00:00 window: 10 artifacts computable at L0, 11 at L1 (including flow duration, directional asymmetry, and short-lived flow patterns), and 9 at L2.

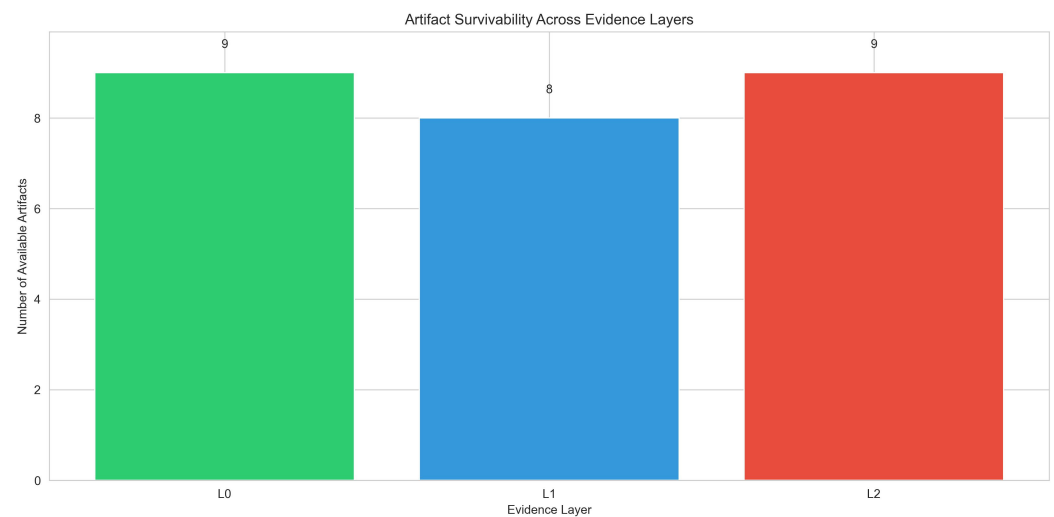


Figure A4. Artifact survivability across evidence layers for the 12:00 time window. The identical pattern (10/11/9) across all three windows demonstrates structural invariance of artifact computability under evidence abstraction.

Appendix C.2. Inference Coverage (Ambiguity Index)

Figures A5 and A6 visualize the ambiguity index (number of supportable ATT&CK tactics) for the 06:00 and 12:00 windows. The monotonic decrease ($9 \rightarrow 8 \rightarrow 7$ tactics) matches the pattern reported in the main text, confirming that inference coverage loss is consistent across traffic conditions.

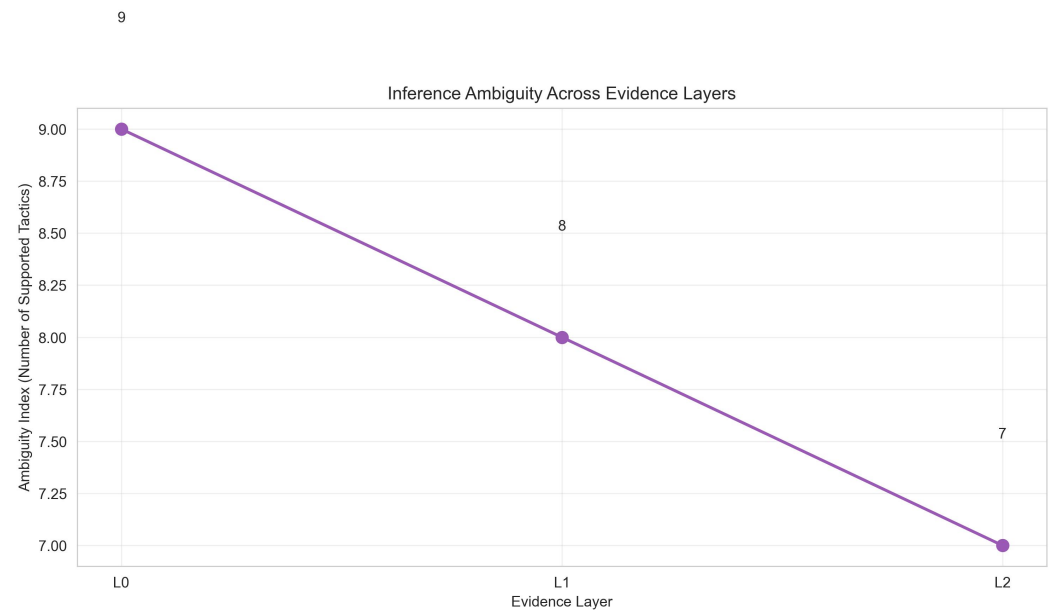


Figure A5. Inference coverage (ambiguity index) for the 06:00 time window, showing the number of ATT&CK tactics that remain supportable at each evidence layer. The monotonic decrease ($9 \rightarrow 8 \rightarrow 7$) matches the pattern observed in the 00:00 window.

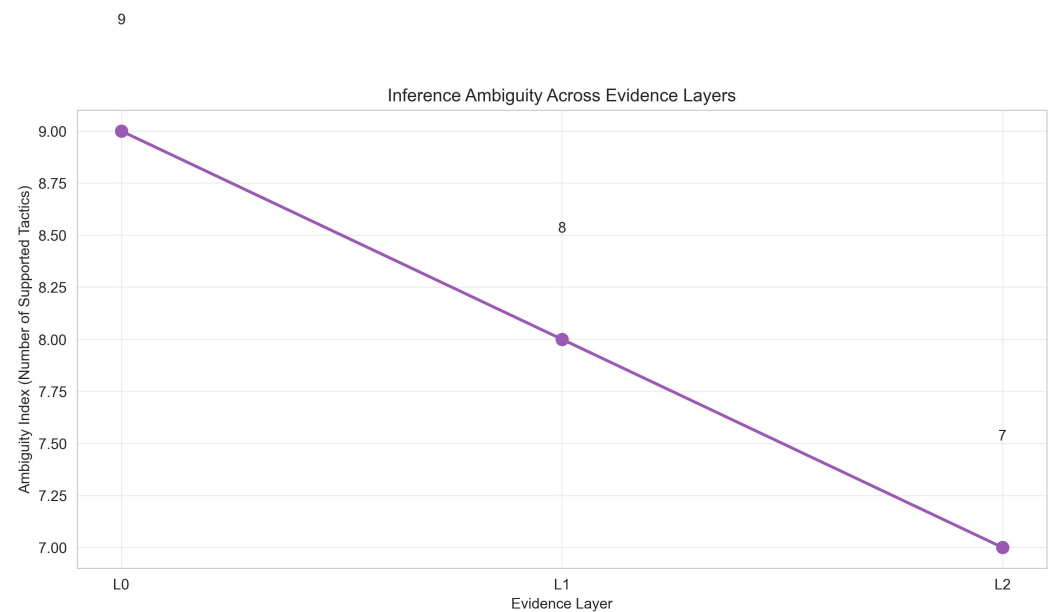


Figure A6. Inference coverage (ambiguity index) for the 12:00 time window. The identical pattern across all three windows ($9/8/7$ tactics) confirms that inference collapse is a structural property of evidence abstraction, independent of traffic characteristics.

Appendix C.3. D3FEND Applicability

Figures A7 and A8 show D3FEND defensive technique category applicability for the 06:00 and 12:00 windows. The non-monotonic pattern ($7 \rightarrow 8 \rightarrow 7$ categories) reflects the transformation from behavioral monitoring (L0) to flow-based controls (L1), followed by the loss of entity-aware anomaly detection (L2), consistent with the main text findings.

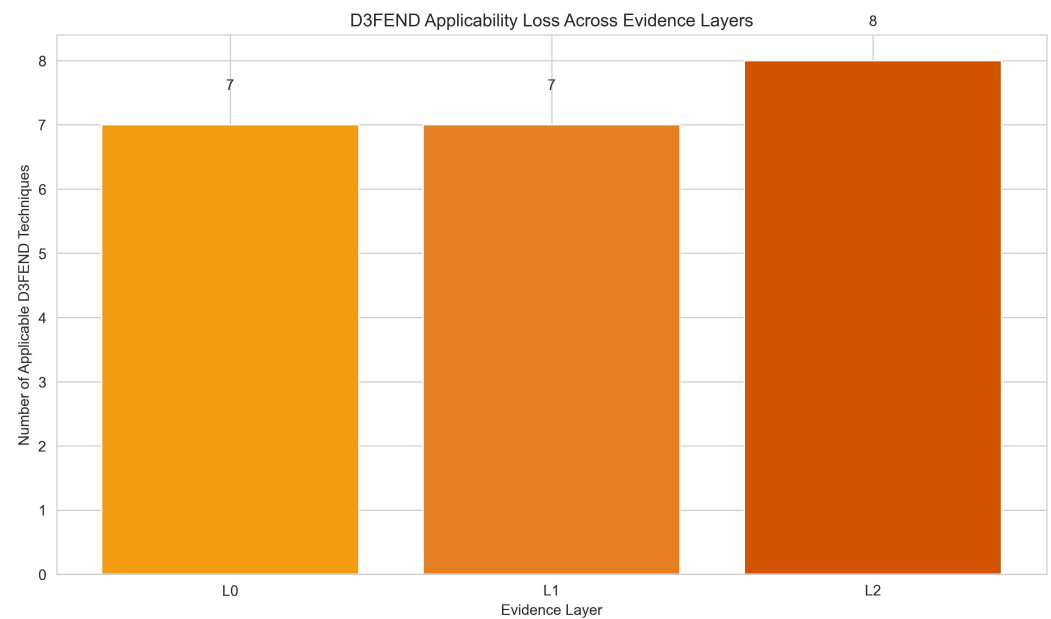


Figure A7. D3FEND defensive technique category applicability for the 06:00 time window. The increase at L1 (7 → 8) reflects the emergence of Flow-based Monitoring, while the decrease at L2 (8 → 7) reflects the loss of Anomaly Detection capabilities requiring per-entity structural information.

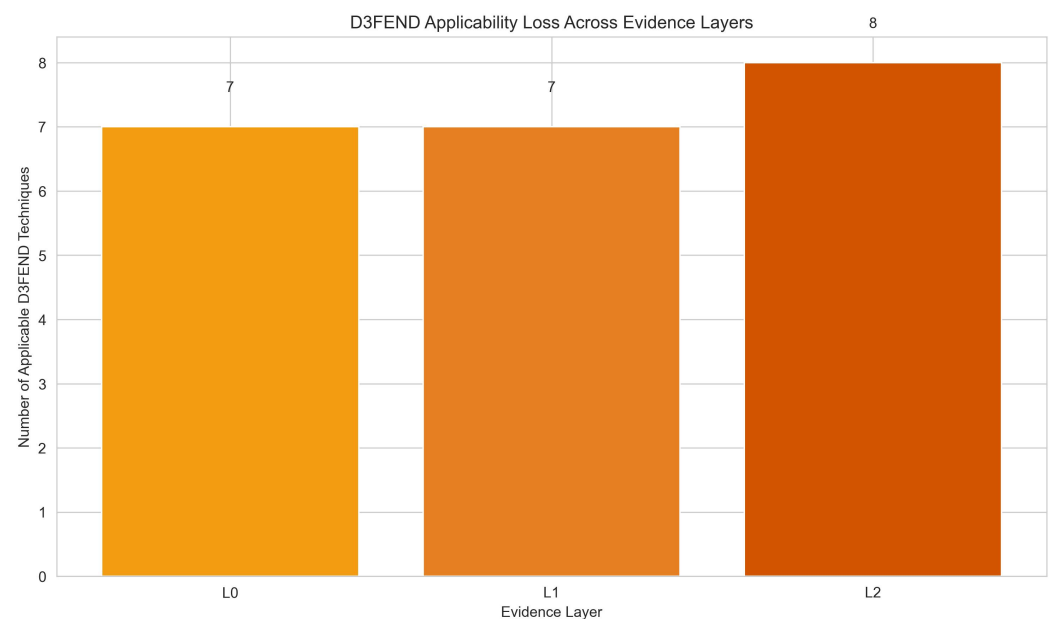


Figure A8. D3FEND defensive technique category applicability for the 12:00 time window. The consistent pattern (7/8/7 categories) across all three windows indicates that defensive applicability transformation is representation-driven rather than traffic-dependent.

References

1. Trevisan, M.; Giordano, D.; Drago, I.; Mellia, M.; Munafo, M. Five years at the edge: Watching internet from the isp network. In *CoNEXT '18: Proceedings of the 14th International Conference on emerging Networking Experiments and Technologies*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–12.
2. Zargar, S.T.; Joshi, J.; Tipper, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2046–2069. [[CrossRef](#)]
3. Velan, P.; Jirsik, T. On the impact of flow monitoring configuration. In *NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium*; IEEE: New York, NY, USA, 2020; pp. 1–7.
4. Zhou, J.; Fu, W.; Hu, W.; Sun, Z.; He, T.; Zhang, Z. Challenges and advances in analyzing tls 1.3-encrypted traffic: A comprehensive survey. *Electronics* **2024**, *13*, 4000. [[CrossRef](#)]

5. European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2025. 2025. Available online: <https://www.enisa.europa.eu> (accessed on 5 December 2025).
6. García, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. *Comput. Secur.* **2014**, *45*, 100–123. [CrossRef]
7. Zeidanloo, H.R.; Amoli, P.V.; Tajpour, A.; Shojaei, M.J.; Zabihi, M.; Kharrazi, M. Botnet detection based on network behavior analysis and flow intervals. *arXiv* **2010**, arXiv:1004.4566.
8. Group, M.W. MAWI Working Group Traffic Archive. 2025. Available online: <https://mawi.wide.ad.jp/mawi/> (accessed on 5 December 2025).
9. Benes, T.; Pesek, J.; Cejka, T. Look at my network: An insight into the ISP backbone traffic. In Proceedings of the 2023 19th International Conference on Network and Service Management (CNSM), Niagara Falls, ON, Canada, 30 October–2 November 2023; pp. 1–7.
10. Maghsoudlou, A.; Gasser, O.; Feldmann, A. Zeroing in on port 0 traffic in the wild. In *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, 29 March–1 April 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 547–563.
11. Gigis, P.; Handley, M.J.; Vissicchio, S. Bad Packets Come Back, Worse Ones Don't. In *ACM SIGCOMM '24: Proceedings of the ACM SIGCOMM 2024 Conference*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 311–326.
12. Hynek, K.; Luxemburk, J.; Pešek, J.; Čejka, T.; Šiška, P. CESNET-TLS-Year22: A year-spanning TLS network traffic dataset from backbone lines. *Sci. Data* **2024**, *11*, 1156. [CrossRef] [PubMed]
13. Schou, M.K.; Poese, I.; Srba, J. Measurement-Noise Filtering for Automatic Discovery of Flow Splitting Ratios in ISP Networks. *Form. Asp. Comput.* **2024**, *36*, 1–18. [CrossRef]
14. Saidi, S.J.; Maghsoudlou, A.; Foucard, D.; Smaragdakis, G.; Poese, I.; Feldmann, A. Exploring network-wide flow data with Flowyager. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1988–2006. [CrossRef]
15. Bühler, T.; Jacob, R.; Poese, I.; Vanbever, L. Enhancing global network monitoring with magnifier. In *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*; USENIX Association: Berkeley, CA, USA, 2023; pp. 1521–1539.
16. Du, Y.; Huang, H.; Sun, Y.E.; Chen, S.; Gao, G. Self-adaptive sampling for network traffic measurement. In *IEEE INFOCOM 2021—IEEE Conference on Computer Communications*; IEEE: New York, NY, USA, 2021; pp. 1–10.
17. He, X.; Xie, X.; Wang, X.; Zhang, L.; Xie, K.; Chen, L.; Cui, Y. FlowSentry: Accelerating NetFlow-based DDoS Detection. In *CCS '25: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 1589–1603.
18. Landau-Feibish, S.; Liu, Z.; Rexford, J. Compact Data Structures for Network Telemetry. *ACM Comput. Surv.* **2025**, *57*, 1–31. [CrossRef]
19. Sun, H.; Li, J.; He, J.; Gui, J.; Huang, Q. Omniwindow: A general and efficient window mechanism framework for network telemetry. In *ACM SIGCOMM '23: Proceedings of the ACM SIGCOMM 2023 Conference*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 867–880.
20. Namkung, H.; Kim, D.; Liu, Z.; Sekar, V.; Steenkiste, P. Telemetry retrieval inaccuracy in programmable switches: Analysis and recommendations. In *SOSR '21: Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 176–182.
21. Srivastava, M.; Hung, S.T.; Namkung, H.; Lin, K.C.J.; Liu, Z.; Sekar, V. Raising the level of abstraction for sketch-based network telemetry with SketchPlan. In *IMC '24: Proceedings of the 2024 ACM on Internet Measurement Conference*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 651–658.
22. Sun, H.; Huang, Q.; Sun, J.; Wang, W.; Li, J.; Li, F.; Bao, Y.; Yao, X.; Zhang, G. AutoSketch: Automatic Sketch-Oriented compiler for query-driven network telemetry. In *NSDI'24: Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*; USENIX Association: Berkeley, CA, USA, 2024; pp. 1551–1572.
23. Diamant, J.; Landau Feibish, S. SetD4: Sets with deletions and decay in the data plane. *Proc. ACM Netw.* **2024**, *2*, 1–22. [CrossRef]
24. Feng, W.; Gao, J.; Chen, X.; Antichi, G.; Basat, R.B.; Shao, M.M.; Zhang, Y.; Yu, M. F3: Fast and Flexible Network Telemetry with an FPGA coprocessor. *Proc. ACM Netw.* **2024**, *2*, 1–22. [CrossRef]
25. Liu, Z.; Namkung, H.; Agarwal, A.; Manousis, A.; Steenkiste, P.; Seshan, S.; Sekar, V. Sketchy with a Chance of Adoption: Can Sketch-Based Telemetry Be Ready for Prime Time? In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 9–16.
26. Koumar, J.; Hynek, K.; Čejka, T.; Šiška, P. CESNET-TimeSeries24: Time Series Dataset for Network Traffic Anomaly Detection and Forecasting. *Sci. Data* **2025**, *12*, 338. [CrossRef]
27. Zhang, T.; Linguaglossa, L.; Gallo, M.; Giaccone, P.; Rossi, D. FloWatcher-DPDK: Lightweight line-rate flow-level monitoring in software. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 1143–1156. [CrossRef]
28. Tangari, G.; Charalambides, M.; Tuncer, D.; Pavlou, G. Accuracy-aware adaptive traffic monitoring for software dataplanes. *IEEE/ACM Trans. Netw.* **2020**, *28*, 986–1001. [CrossRef]

29. Wang, Y.; Wang, X.; Xu, S.; He, C.; Zhang, Y.; Ren, J.; Yu, S. FlexMon: A flexible and fine-grained traffic monitor for programmable networks. *J. Netw. Comput. Appl.* **2022**, *201*, 103344. [[CrossRef](#)]
30. Sha, M.; Guo, Z.; Wang, K.; Zeng, X. A high-performance and accurate FPGA-based flow monitor for 100 Gbps networks. *Electronics* **2022**, *11*, 1976. [[CrossRef](#)]
31. Doriguzzi-Corin, R.; Knob, L.A.D.; Mendozzi, L.; Siracusa, D.; Savi, M. Introducing packet-level analysis in programmable data planes to advance network intrusion detection. *Comput. Netw.* **2024**, *239*, 110162. [[CrossRef](#)]
32. Fink, I.B.; Kunze, I.; Hein, P.; Pennekamp, J.; Standaert, B.; Wehrle, K.; R  th, J. Advancing Network Monitoring with Packet-Level Records and Selective Flow Aggregation. In *NOMS 2025—2025 IEEE Network Operations and Management Symposium*; IEEE: New York, NY, USA, 2025.
33. Hardegen, C. Scope-based flow monitoring to improve traffic analysis in programmable networks. In *Proceedings of the 2022 18th International Conference on Network and Service Management (CNSM)*, Thessaloniki, Greece, 31 October–4 November 2022; pp. 254–260.
34. Papadogiannaki, E.; Ioannidis, S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [[CrossRef](#)]
35. Sharma, A.; Lashkari, A.H. A survey on encrypted network traffic: A comprehensive survey of identification/classification techniques, challenges, and future directions. *Comput. Netw.* **2025**, *257*, 110984. [[CrossRef](#)]
36. Sarhan, S.A.E.; Youness, H.A.; Bahaa-Eldin, A.M. A framework for digital forensics of encrypted real-time network traffic, instant messaging, and VoIP application case study. *Ain Shams Eng. J.* **2023**, *14*, 102069. [[CrossRef](#)]
37. Al-Sada, B.; Sadighian, A.; Oligeri, G. MITRE ATT&CK: State of the art and way forward. *ACM Comput. Surv.* **2024**, *57*, 1–37. [[CrossRef](#)]
38. Yousaf, A.; Zhou, J. From sinking to saving: MITRE ATT &CK and D3FEND frameworks for maritime cybersecurity. *Int. J. Inf. Secur.* **2024**, *23*, 1603–1618. [[CrossRef](#)]
39. Vaseghipanah, M.; Jabbehdari, S.; Navidi, H. A Game-Theoretic Approach for Quantification of Strategic Behaviors in Digital Forensic Readiness. *J. Cybersecur. Priv.* **2025**, *5*, 105. [[CrossRef](#)]
40. MITRE Corporation. MITRE D3FEND: A Knowledge Graph of Cybersecurity Countermeasures. 2025. Available online: <https://d3fend.mitre.org/> (accessed on 5 December 2025).
41. Inacio, C.M.; Trammell, B. YAF: Yet another flowmeter. In *LISA'10: Proceedings of the 24th International Conference on Large Installation System Administration*; USENIX Association: Berkeley, CA, USA, 2010.

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.