



Threat Intelligence for IoT Cybersecurity: Frameworks, Detection Methodologies, and Deployment Strategies

Arunabh K S^a, Ms. Riya K Prakash^b, Dr. L C Manikandan^{c*}

^a B.Tech Student, Department of CSE, Universal Engineering College, Thrissur, Kerala, India. E-mail: arunabhkshiju@gmail.com

^b Assistant Professor, Department of CSE, Universal Engineering College, Thrissur, Kerala, India. E-mail: riyajothish13@gmail.com

^c Professor, Department of CSE, Universal Engineering College, Thrissur, Kerala, India. E-mail: lmanikandan@gmail.com

ABSTRACT :

For much of the past two decades, organizations defended their digital assets using static, signature-driven tools anchored at a well-defined network boundary. That assumption is no longer tenable. The rapid, uncontrolled expansion of Internet of Things (IoT) devices spanning consumer gadgets, industrial sensors, and embedded medical equipment has created a sprawling, largely unmanaged attack surface that legacy boundary defenses were never designed to observe, let alone protect. Constrained processing budgets, fragmented protocol stacks (MQTT, CoAP, Zigbee, Modbus), infrequent firmware updates, and always-on connectivity combine to make IoT fleets an attractive staging ground for botnets, ransomware pivots, and covert reconnaissance. Threat Intelligence (TI), when purpose-built for these constraints, offers a way out of this impasse: it converts scattered, low-context telemetry into structured, actionable knowledge that machines and analysts alike can act upon before damage occurs. This paper examines how threat intelligence can be engineered specifically for IoT cybersecurity, tracing the full operational path from constrained-device data capture and on-device inference through standardized indicator exchange and cooperative, privacy-conscious intelligence sharing across organizational boundaries. Building on this foundation, the paper surveys five representative deployment contexts—industrial cyber-physical systems, consumer smart-home networks, malware-focused honeypot and sandbox pipelines, semantically aware automated response engines, and the emerging domains of smart agriculture and connected healthcare—to illustrate how the same underlying intelligence lifecycle adapts to very different risk profiles and latency requirements. A comparative assessment of the enabling techniques and a discussion of practical deployment frictions round out the analysis before the paper closes with open research directions and a synthesis of its findings.

Keywords: Threat Intelligence (TI), Internet of Things (IoT) Security, Edge Analytics, TinyML, Federated Learning, STIX/TAXII, Decentralized Intelligence Sharing, Industrial IoT (IIoT), Honeypots, Semantic Threat Detection.

1. Introduction

Enterprise network defense was, for most of its history, organized around a single architectural premise: protect the boundary, and the interior takes care of itself. Firewalls, gateway-level intrusion detection, and centralized log aggregation were placed at the few points where an organization's network met the outside world, on the assumption that everything behind those chokepoints could be observed and controlled. That premise depended on infrastructure staying put—on servers, workstations, and applications living inside a small number of physical sites that a security team could realistically instrument.

The Internet of Things has quietly dismantled that premise. Manufacturing floors, hospitals, retail chains, utilities, and ordinary households now run on constellations of small, purpose-built devices whose security posture a centralized team can no longer directly instrument. Three structural properties of IoT deployments make this shift consequential. First, protocol diversity: devices exchange telemetry and commands over lightweight application-layer protocols (MQTT, CoAP), industrial fieldbus protocols (Modbus, BACnet), and low-power radio standards (Zigbee, LoRaWAN, Bluetooth Low Energy) that differ substantially from the HTTP/TLS traffic that conventional security appliances are tuned to inspect [3], [6]. Second, resource constraints: the majority of IoT endpoints run on microcontrollers with kilobytes, not gigabytes, of memory, and cannot support full endpoint-detection agents, deep packet inspection, or standard public-key cryptographic handshakes without unacceptable latency or battery drain [2]. Third, persistent and often unattended connectivity: many devices remain powered on and network-reachable indefinitely, with infrequent firmware updates, creating a durable population of exploitable, internet-facing endpoints [5].

Adversaries have adapted quickly to this landscape. Botnets such as Mirai and its many derivatives have demonstrated that default credentials and unpatched firmware across millions of consumer devices can be aggregated into attack infrastructure capable of record-scale distributed denial-of-service campaigns [6]. In industrial settings, attackers increasingly treat compromised IoT or operational-technology (OT) endpoints as a foothold for lateral movement toward programmable logic controllers (PLCs) and supervisory control systems, where the consequences of compromise extend from data loss into physical safety incidents [7], [11].

Threat Intelligence for IoT Cybersecurity responds to this landscape by reframing defense as a continuous, evidence-driven process rather than a static rule set. Rather than waiting for a signature update, an intelligence-led architecture ingests telemetry from sensors, gateways, honeypots, and external

feeds; extracts indicators of compromise and adversary tactics, techniques, and procedures; correlates them against known threat behavior; and pushes enforcement decisions back to the network edge with minimal delay [1], [3]. This paper surveys the architectural components that make this loop practical inside resource-constrained IoT environments, and examines how the resulting capability is applied across a range of real-world deployment contexts.

The remainder of the paper is organized as follows. Section 2 sets out the architecture and core operational layers of an IoT-oriented threat intelligence pipeline. Section 3 details five representative application domains. Section 4 offers a comparative assessment of the enabling techniques discussed. Section 5 discusses implementation challenges and operational trade-offs. Section 6 outlines open research directions, and Section 7 concludes.

2. Architecture of an IoT Threat Intelligence Pipeline

Building a functioning threat intelligence capability for IoT environments requires an integrated pipeline that can absorb high-velocity, heterogeneous telemetry at the edge, extract meaningful features under tight resource budgets, express the resulting indicators in a form that other systems and organizations can consume, and coordinate a defensive response across many independently owned devices and networks [5]. The architecture examined in this paper is organized into four cooperating layers.

2.1. Telemetry Acquisition and Normalization

An intelligence pipeline is only as good as the telemetry that feeds it, and IoT environments complicate collection in ways that conventional endpoint security does not encounter. Passive capture of IoT-specific traffic—MQTT topic publications, CoAP resource requests, Modbus register operations—allows a monitoring system to flag protocol misuse without touching the device itself [12]. For the subset of devices capable of hosting local instrumentation, system-call traces and kernel logs provide additional visibility into unauthorized code execution or firmware tampering, while low- and high-interaction honeypots that imitate common IoT weaknesses give defenders a controlled way to observe how automated scanners and emerging botnets actually behave [1]. Whatever the source, raw captures are normalized: redundant payload bytes are stripped, timestamps and field encodings are reconciled across protocols, and each event is reduced to a compact feature vector suitable for downstream inference.

2.2. Edge-Resident Analysis: TinyML and On-Device Inference

Backhauling raw telemetry from thousands of distributed endpoints to a centralized analytics platform consumes bandwidth those endpoints do not have and introduces latency that a real-time safety system cannot tolerate. Tiny Machine Learning (TinyML) addresses this by moving inference to the device or gateway itself: quantized neural networks, pruned decision trees, and lightweight support-vector classifiers are compressed until they fit within a few hundred kilobytes of memory, allowing a microcontroller to score incoming packet bursts or sensor readings in real time [2]. Rather than exporting raw data, the device exports a small, already-interpreted verdict, which dramatically reduces both the bandwidth footprint and the time-to-detection of the overall system.

2.3. Structured Representation: STIX/TAXII and Correlation

A detection is only actionable intelligence once it can be understood and consumed outside the system that produced it. Structured Threat Information eXpression (STIX 2.1) defines a JSON-based vocabulary of domain objects—Indicator, Malware, Attack-Pattern, Observed-Data, Identity—that describe a threat in a form both humans and machines can parse consistently, while Trusted Automated eXchange of Intelligence Information (TAXII 2.1) is the companion RESTful transport used to publish and subscribe to that content in close to real time [3]. A correlation engine ingests both externally sourced STIX feeds and internally generated observations, typically using a graph data model, and cross-references local anomalies against the MITRE ATT&CK matrix for ICS and IoT to distinguish ordinary network jitter from the early stages of a recognizable intrusion pattern.

2.4. Decentralized Sharing: Federated Learning and Distributed Ledgers

Centralizing threat intelligence in a single repository creates a single point of failure and asks participating organizations to trust a third party with sensitive telemetry. Decentralized architectures address both concerns by pairing Federated Learning (FL) with blockchain-anchored consensus: each participating site trains an anomaly-detection model on its own private telemetry and submits only the resulting model weights, not the underlying logs, to a shared ledger [8]. A consensus mechanism, whether Proof of Authority or a Byzantine-fault-tolerant smart contract, checks the validity of each submission before folding it into a global model [9], [10]. The outcome is a detection capability that improves with every participant's experience, including newly observed zero-day patterns, without any single organization ever exposing its raw internal data to the others.

3. Deployment Scenarios and Applications

The layered pipeline described in Section 2 is deliberately general; its value is realized only once adapted to the operational realities of a specific deployment context. The following subsections examine five representative domains.

3.1. Industrial IoT and Cyber-Physical Systems

Industrial IoT (IIoT) deployments—power distribution substations, manufacturing lines, water treatment facilities—differ from consumer IoT in one critical respect: a malicious command does not merely leak data; it can alter a physical process. An unauthorized Modbus register write to a pump controller, for instance, can translate directly into equipment damage, product loss, or a safety incident [7].

Context-aware CTI platforms for IIoT, exemplified by architectures such as SmartSentry, combine network-layer packet inspection with operational-technology telemetry to construct behavioral baselines for each programmable logic controller (PLC) and remote terminal unit (RTU) on the network [11]. When an incoming command deviates from this baseline—an out-of-sequence register write, an unexpected source address, an operation outside the normal duty cycle—the platform estimates the physical-process risk associated with the deviation and can trigger automated micro-segmentation at the software-defined networking (SDN) layer, isolating the affected controller before the command can execute, rather than merely logging the event after the fact.

3.2. Dynamic Risk Assessment in Smart Home Environments

Consumer smart-home environments present a different problem profile: dozens of heterogeneous devices—locks, cameras, thermostats, voice assistants—are managed by residents with no security training, and the device population changes continuously as products are added, replaced, or updated. Static, one-time risk assessments quickly become stale in this setting [4].

Dynamic risk-assessment frameworks address this by continuously re-scoring each device's exposure as conditions change. Live threat feeds, consumed through open platforms such as OpenCTI, are matched against behavioral profiles built for each local device. When a device deviates from its expected behavior—for example, a smart television initiating outbound connections to an address associated with known command-and-control infrastructure—the framework computes a risk score from the estimated likelihood of compromise and the operational impact of the affected device. Lower-impact devices can have mitigating firewall rules applied automatically; higher-impact devices, such as door locks or medical alert systems, instead surface a clear, human-readable recommendation so the resident can decide how to respond [4].

3.3. IoT Malware Analysis, Reverse Engineering, and Honeygot Feeds

IoT-specific malware families—Mirai, BASHLITE, and their many derivatives—target an unusually wide range of processor architectures (ARM, MIPS, PowerPC, and legacy x86 variants found in older gateways), which limits the usefulness of desktop-oriented malware analysis tooling built primarily for x86 [6].

A dedicated collection and analysis pipeline typically begins with decoy services that emulate commonly exploited entry points—open Telnet with factory-default credentials, vulnerable UPnP implementations, exposed MQTT brokers—to capture live malware samples as they attempt to spread. Captured binaries are then executed inside isolated, architecture-matched sandboxes, commonly built on QEMU-based cross-architecture emulation, while system calls, file-system modifications, and outbound network connections are logged in detail. Automated extraction routines mine this execution log for command-and-control IP addresses and domains, dynamic-DNS patterns, and cryptographic hashes, and package the results as STIX objects that can be pushed to network gateways to block further infection attempts across the wider install base [1], [6].

3.4. Deep Learning-Based Semantic Reasoning and Automated Edge Response

Pattern-matching against known indicators catches previously observed threats efficiently, but it says little about intent. Semantic reasoning engines extend the pipeline by applying natural-language and graph-based deep-learning models—transformer architectures and graph neural networks—to unstructured inputs such as OSINT threat reports, vendor advisories, and raw packet headers [14]. By building semantic relationships between low-level network observations and higher-level adversary objectives, these engines can distinguish, for example, a reconnaissance sweep intended to map the network from a payload staged for data exfiltration, or an attack aimed at simple device bricking. Once intent is estimated, the engine can compile and deploy a proportionate, automated response—a localized eBPF packet filter, an OpenFlow rule pushed to an SDN controller, or a temporary access-control change—tuned to the specific behavior observed rather than a generic block rule [14].

3.5. Emerging Domains: Smart Agriculture and Connected Healthcare

Two further domains illustrate how the same architectural pattern generalizes beyond the industrial and consumer settings already discussed. Precision agriculture deployments increasingly rely on networks of soil, irrigation, and livestock-monitoring sensors spread across large, often unattended physical areas with intermittent connectivity; threat intelligence platforms adapted to this setting must tolerate long communication gaps, prioritize battery-preserving sampling, and re-synchronize risk scores in bulk once connectivity is restored, since continuous real-time reporting is often not physically possible for field-deployed sensors [13].

Connected healthcare presents the opposite constraint profile: bandwidth and power are less scarce, but the tolerance for false positives and for any data exposure is far lower, since a misclassified anomaly on an infusion pump or a cardiac monitor can directly affect patient safety, and any indicator-sharing mechanism must be compatible with health-data privacy regulation. In both domains, the underlying four-layer pipeline from Section 2 remains unchanged; what differs is the tuning of sampling intervals, confidence thresholds, and sharing scope to match the domain's specific risk and connectivity profile [15].

4. Comparative Assessment of Enabling Techniques

Table 1 summarizes the operational trade-offs of the principal techniques discussed in Sections 2 and 3, providing a practical reference for architects selecting an appropriate combination for a given deployment [15].

Table 1 - Comparative Assessment of IoT Threat-Intelligence Enabling Techniques.

Technique	Primary Strength	Primary Limitation / Best-Fit
TinyML edge inference [2]	Low latency; minimal bandwidth use; preserves data locality	Constrained model accuracy vs full ML. Best-Fit: Battery-powered sensors, remote gateways.
STIX/TAXII indicator exchange [3]	Standardized, machine-readable, cross-organization sharing	Requires disciplined data modeling. Best-Fit: Cross-enterprise & public feeds.
Federated learning [10]	Collaborative model improvement without raw data sharing	Sensitive to poisoned updates. Best-Fit: Multi-tenant or multi-site deployments.
Blockchain consensus [8], [9]	Tamper-evident, auditable model-update history	Consensus latency; storage overhead. Best-Fit: High-trust, multi-party federations.
Semantic deep learning [14]	Captures adversary intent, not just indicator matches	Higher compute cost; needs quality data. Best-Fit: SOC-grade automated engines.
Honeypot-driven malware capture [1], [6]	Early visibility into novel, in-the-wild payloads	Limited to active decoys. Best-Fit: Perimeter-facing gateways, ISP sensors.

5. Implementation Challenges and Operational Trade-Offs

Operational challenges in deploying IoT threat intelligence at scale encompass three major domains:

5.1. Hardware and Resource Heterogeneity

A large share of installed IoT endpoints, particularly legacy sensors, simply lack the processing headroom, memory, or storage to run local cryptographic routines or even a lightweight inference model. Where this is the case, architects must offload telemetry processing to a nearby gateway acting on the constrained device's behalf, at the cost of an additional trust boundary [2], [15].

5.2. Telemetry Overhead and Network Congestion

Continuous, verbose logging across thousands of low-power wireless nodes—Zigbee, LoRaWAN, and similar radio standards—can saturate shared spectrum and shorten battery life well below expected service intervals. Dynamic sampling, event-triggered reporting, and on-device feature extraction are necessary to keep transmission volume within what the underlying radio and power budget can sustain [12].

5.3. False-Positive Management in Critical Operations

In industrial and healthcare settings, an automated response triggered by an inaccurate risk score can itself cause an outage or a safety event, arguably worse than the threat it was meant to prevent. Production deployments therefore favor conservative confidence thresholds, staged responses that start with alerting before escalating to isolation, and human-in-the-loop confirmation for any action affecting high-criticality assets [7], [15].

6. Conclusion

The perimeter-oriented security model that served centralized enterprise networks for two decades does not transfer cleanly to environments built from billions of heterogeneous, resource-constrained, and persistently connected devices. Threat Intelligence for IoT Cybersecurity responds to this gap by treating defense as a continuous pipeline: telemetry is collected close to its source, distilled through on-device TinyML inference, expressed through standardized STIX/TAXII indicators, correlated against known adversary behavior, and, where appropriate, shared across organizational boundaries through privacy-preserving federated and blockchain mechanisms.

Applied across industrial control systems, smart-home ecosystems, malware-research pipelines, semantic response engines, and emerging domains such as agriculture and healthcare, this architecture demonstrates that intelligence-led defense can operate at the low latency and limited resource budget that IoT environments demand. The remaining obstacles—hardware heterogeneity, telemetry overhead, and the risk of false-positive-driven disruption—are operational rather than conceptual, and the trajectory of current research suggests they will continue to narrow as edge hardware, standardized indicator formats, and decentralized sharing mechanisms mature together.

Acknowledgements

The authors express their gratitude to the Department of Computer Science & Engineering at Universal Engineering College, Thrissur, Kerala, India, for providing research facilities and technical guidance throughout this investigation.

REFERENCES

- [1] D. Mendez Mena and B. Yang, "Decentralized Actionable Cyber Threat Intelligence for Networks and the Internet of Things," *IoT*, vol. 2, no. 1, pp. 1-16, 2020.
- [2] A. Dutta and S. Kant, "Implementation of a Cyber Threat Intelligence Platform on IoT Using a TinyML Approach for Deceiving Cyber Invasion," *Proc. Int. Conf. on Artificial Intelligence and Smart Systems*, 2021.
- [3] S. Brown, et al., "Cyber Threat Intelligence for IoT Using Machine Learning," *IEEE Trans. Network and Service Management*, vol. 19, no. 3, pp. 2100-2114, 2022.
- [4] A. Collen and N. A. Nijdam, "Can I Sleep Safely in My Smart Home? A Novel Framework on Automating Dynamic Risk Assessment in IoT Environments," *Electronics*, vol. 11, no. 7, p. 1123, 2022.
- [5] R. Alvarez, et al., "ESSecA: An Automated Expert System for Threat Modelling and Penetration Testing for IoT Ecosystems," *Computers & Security*, vol. 118, p. 102730, 2022.
- [6] P. Kumar, et al., "Tools and Techniques for Collection and Analysis of Internet-of-Things Malware: A Systematic State-of-the-Art Review," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1-36, 2022.
- [7] A. Sharma, et al., "An Intelligent Context-Aware Threat Detection and Response Model for Smart Cyber-Physical Systems," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10450-10462, 2023.
- [8] P. T. Duy, et al., "FedChain-Hunter: A Reliable and Privacy-Preserving Aggregation for Federated Threat Hunting in SDN-Based IIoT," *Internet of Things*, vol. 24, p. 100966, 2023.
- [9] M. Hassan, et al., "Collaborative Threat Intelligence: Enhancing IoT Security Through Blockchain and Machine Learning Integration," *Journal of Network and Computer Applications*, vol. 221, p. 103780, 2024.
- [10] L. Zhao, et al., "FedCTI: Federated Learning and Cyber Threat Intelligence on the Edge for Secure IoT Networks," *IEEE Trans. Information Forensics and Security*, vol. 19, pp. 3200-3215, 2024.
- [11] K. Patel, et al., "SmartSentry: Cyber Threat Intelligence in Industrial IoT," *IEEE Micro*, vol. 44, no. 2, pp. 45-54, 2024.
- [12] V. Singh, et al., "Cyber Attack Detection in IoT-WSN Devices with Threat Intelligence Using Hidden and Connected Layer Based Architectures," *Wireless Networks*, vol. 30, no. 4, pp. 1820-1835, 2024.
- [13] I. Ali, et al., "A Novel Cyber Threat Intelligence Platform for Evaluating the Risk Associated with Smart Agriculture," *Computers and Electronics in Agriculture*, vol. 228, p. 108500, 2025.
- [14] F. Rahman, et al., "A Computational Framework for IoT Security Integrating Deep Learning-Based Semantic Algorithms for Real-Time Threat Response," *Future Generation Computer Systems*, vol. 162, p. 107400, 2025.
- [15] A. Al-Sulaihi, et al., "Toward Using Cyber Threat Intelligence with Machine and Deep Learning for IoT Security: A Comprehensive Study," *IEEE Access*, vol. 13, pp. 15200-15225, 2025.