Intelligence-driven Contextual Assessment of Risks in Unseen Scenarios (ICARUS)

Ricardo Oliveira
pg54177@alunos.uminho.pt
University of Minho
Braga, Portugal

Abstract

Cyber Threat Intelligence (CTI) provides organisations with valuable information on indicators, attack patterns, and threat actors, but its application in real-time detection pipelines remains limited. Existing SIEM and IDS platforms are usually based on static rules or isolated IOC matching, lacking mechanisms to dynamically contextualise observations with multi-source intelligence. This paper presents an autonomous CTI-driven alert system that integrates external STIX feeds with real-time endpoints and network telemetry to generate contextual security alerts.

The presented system, ICARUS, dynamically correlates local observations with current CTI, assesses risk scores, and activates relevant data collectors to improve detection accuracy. A complete proof of concept was developed and deployed on a controlled test bed to evaluate the system. The results show that CTI-based correlations allow organisations to establish automatic detection mechanisms without the need for manual intervention to identify emerging threats. The results show that CTI can be implemented as a practical, autonomous detection pipeline and has the potential to be a part of the next generation of proactive cyber defence strategies.

Keywords

Cyber Threat Intelligence, Cybersecurity, STIX, osquery

1 Introduction

The complexity and volume of cyber threats exceed the capacity of traditional security monitoring systems. Modern Security Information and Event Management (SIEM) and Intrusion Detection Systems (IDS) are usually based on static rules and signature-based mechanisms, which often leads to limited adaptability to new and evolving threats. Although Cyber Threat Intelligence has emerged as a rich source of practical knowledge, providing Indicators of Compromise (IOCs), attack patterns, and threat actor information, its integration into operational detection pipelines is still limited. CTI is often used for enrichment or manual analysis, preventing organisations from fully exploiting its proactive defence potential.

Previous research has focused on the standardisation and sharing of threat intelligence with standards such as STIX and TAXII, but there is still a clear gap: despite the volume of threat intelligence shared, no system correlates it with local telemetry data to autonomously generate real-time contextual alerts to previously unseen threats. As a result, and despite the latest machine learning and artificial intelligence advancements, organisations are dependent on intelligence workflows that tend to be reactive or anomaly-based, lacking the ability to proactively, and automatically, identify new threats based on the latest intelligence.

This paper addresses this gap by presenting a CTI-based alerting system called ICARUS (Intelligence-driven Contextual Assessment of Risks in Unseen Scenarios), which incorporates modern CTI standards into dynamic detection pipelines. The system ingests external sources of information, linking them to local observations such as processes and network connections, resulting in autonomous and contextual alerts. When new threats are detected, specialised data collectors can be activated to improve local visibility and further improve the detection capabilities of the system.

This work makes four key contributions:

- The design of a modular architecture that integrates real-time telemetry and structured CTI to enable autonomous and contextual alerts for emerging threats;
- The implementation of a complete proof of concept that is capable of ingesting STIX feeds, correlating external observations with the local threat landscape;
- The illustration of how dynamic CTI correlations can enable early detection of emerging threats and support proactive defence strategies.

This paper views CTI-driven alerting as a viable approach to strengthening modern cyber security infrastructures, moving more efforts toward a proactive and intelligence-driven defence posture.

The remainder of this paper is organised as follows: Section 2 reviews related work in the field of CTI and threat detection. Section 3 discusses the current limitations in existing systems. Section 4 presents the architecture of the proposed system. Section 5 details the implementation of the system. Section 6 evaluates the system's performance, and Section 7 concludes the paper with final remarks and future work directions.

2 Related Work

CTI has developed to be a pivotal element of the current cybersecurity landscape, underpinned by ongoing standardisation initiatives and the deployment of dedicated information-sharing infrastructures. The Structured Threat Information eXpression (STIX) language [1, 13] provides a comprehensive and extensible schema for the formalised representation of indicators of compromise, attack patterns, and threat actor behaviours. It separates itself from complementary formats such as CyBOX [2], Common Vulnerabilities and Exposures (CVE) [17], and specification frameworks including the Common Vulnerability Reporting Framework (CVRF) and its successor, the Common Security Advisory Framework (CSAF) [10, 19], by its capability to model a broad spectrum of cyber threat intelligence rather than being restricted to a narrow subset of observables, while remaining both human and machine-interpretable. This, in turn, facilitates more reliable automated exchange and processing of cyber threat and vulnerability information.

In order for these standards to be used, additional intelligence sharing protocols and platforms have been developed. The Trusted Automated eXchange of Indicator Information (TAXII) [5] specifies the primary transport protocol for the exchange of STIX-formatted data, while collaborative platforms such as MISP [23] and AlienVault OTX [15] support the community aspect of threat intelligence sharing. Kampanakis [14] surveys these information-sharing options, highlighting their contribution to improving organisational situational awareness.

In addition, academic research has made advances in the CTI production and enrichment workflows. Sadique et al.[20] introduce methods for the privacy-preserving generation of STIX information, while Iqbal et al.[12] developed automated pipelines for intelligence generation. More recent contributions, such as Stixnet [16] and STIX graph-based IoCs enhancement techniques [3], concentrate on extracting threat information and improving the overall quality of intelligence.

Surveys and analyses provide additional information on the challenges associated with the adoption of cyber threat intelligence in current cybersecurity workflows. Ramsdale et al. [18] examine heterogeneity among CTI formats, while Schlette et al. [21, 22] perform a comprehensive assessment of CTI quality while analysing the maturity of intelligence-driven processes in security operations centres (SOC). More general studies of the intelligence lifecycle [4, 8] highlight persistent obstacles in multi-source data fusion and automation, which align with the challenges reported in CTI-sharing surveys such as [24].

Despite recent advances, existing processes predominantly focus on CTI standardisation, sharing mechanisms and platforms, or its usage to enrich incident response workflows. In practice, current methodologies continue to depend on predefined rules and IOC matching, with ongoing efforts to improve detection with AI and machine learning techniques [9]. Methodologies that integrate CTI more closely into automated detection pipelines, such as CTI-informed incident response frameworks [11], remain largely conceptual or require substantial oversight by human analysts to achieve its effectiveness.

Finally, recent data provide a clear need for innovation as threats discovered continue to increase year after year.

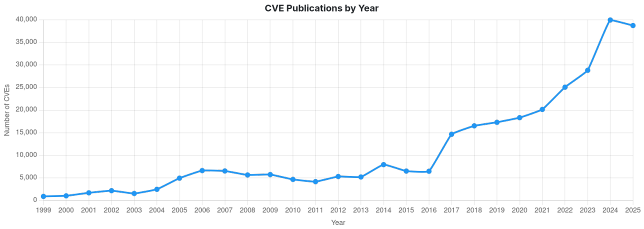

Figure 1: CVE Publications by Year, [6]

As shown in Figure 1, the number of published CVE's is increasing rapidly. With nearly 40000 new vulnerabilities published in 2025 alone [25], the need for rapid adaptability and actionable alerting is more important than ever.

In conclusion, while existing studies lay the foundations for the creation and sharing of CTI among different entities, making CTI actionable for automated alerting and detection remains an open challenge. This shortcoming underscores the need for practical alerting pipelines driven by threat intelligence, which this work aims to address. These limitations are further discussed in Section 3, laying the foundation for the proposed approach.

3 Current Limitations

Although standards and sharing ecosystems for cyber threat intelligence have undergone substantial research over the years [1, 5, 13, 23], their operational integration into the real-time detection process remains limited. In prevailing practice, most SIEM and IDS platforms still utilise CTI as an auxiliary enrichment for information or atomic IOC matching, rather than as a primary driver of detection logic. This under-use fails to take advantage of the extensive reach of contemporary CTI formats both during detection and contextualisation of threats.

A major constraint is the lack of fully autonomous mechanisms capable of correlating local data with multi-source external threat intelligence. Although prior work has demonstrated the feasibility of structured threat representation [2, 10, 19] and automated generation of CTI artefacts [12, 20], these capabilities are rarely incorporated into production-grade cybersecurity systems. This is a crucial gap, as the dynamic nature of cyber threats requires a contextual analysis in real-time that transcends static IOC matching. Not only does this affect the detection of new threats, it also limits the ability to identify complex attack patterns that may span multiple indicators and tactics.

Timeliness constitutes an additional significant challenge. Although CTI platforms such as AlienVault OTX [15] and MISP [23] are continuously updated, operational environments rarely incorporate new indicators available immediately. This temporal delay may constrain the effectiveness of defensive measures, particularly when taking into account the decreasing average time-to-exploit (TTE) observed in contemporary threat landscapes year-over-year [7]. To remedy this, systems must be designed to immediately assimilate and alert upon new intelligence, without necessitating extensive manual intervention.

Finally, related research provides additional evidence that heterogeneous and insufficient data quality [21], as well as inconsistencies among CTI sources [18], significantly hinder the reliability of automated processes and impede large-scale operational integration. The variability in data quality is then a major obstacle, as it complicates both the development of frameworks capable of processing diverse intelligence formats and the establishment of trustworthy alerting mechanisms that can act autonomously. It is therefore imperative to filter and normalise incoming intelligence to ensure its relevance and reliability before it is utilised in detection workflows.

3.1 Main Limitations

The following limitations summarise the structural gaps preventing effective CTI operationalisation:

- **Static IOC matching dominates current systems**, preventing contextual or behaviour-aware detection.
- **Lack of autonomous correlation** between local telemetry and multi-source CTI reduces detection depth.

- **Slow adaptation to newly published indicators** leaves defenders exposed to emerging threats.
- **High variability in quality** of the CTI and the consistency of the format complicates the automated use at scale.
- **Absence of adaptive evidence collection**, causing ambiguous or partial matches to be misclassified.

In summary, although Cyber Threat Intelligence (CTI) provides extensive, structured knowledge about adversarial activities, existing operational platforms are largely inadequate in transforming this information into dynamic, context-aware, and operationally actionable alerts. To mitigate these limitations, the subsequent section presents a modular system architecture specifically designed to operationalise CTI by enabling real-time correlation, adaptive data acquisition, and autonomous alert generation.

4 System Definition

This section presents the proposed system for operationalising CTI by transforming continuously updated threat information into dynamic, contextual, and actionable alerts. The goal is to create a modular, extensible, and scalable architecture that is capable of transforming raw intelligence into actionable alerts in environments where threats evolve rapidly and detection windows are increasingly narrow.

The following sections outline the system design objectives, provide an overview of the core architectural components, and describe the data collection mechanisms that support the correlation process. Finally, the complete pipeline is detailed from ingestion to alert generation, highlighting how each stage contributes to the overall operationalisation of CTI.

4.1 Design Goals

In orderd to design the ICARUS system the technical and operational shortcomings described in section 3 were analysed to derive a set of objectives that the architecture must fulfil. These objectives translate abstract challenges of associated with the operationalisation of CTI into concrete requirements that guide the construction of the solution.

Furthermore, these objectives include the principles necessary for a system capable of resisting the dynamic nature of the modern threat environment. They emphasize the alert relevance, threat contextualisation, and adaptability attributes that are essential for the practical deployment of individuals and organizations seeking to use CTI for proactive and autonomous detection.

With these considerations in mind, the following design goals were established:

- **CTI Ingestion and Normalisation:** ICARUS must seamlessly ingest and normalise both external CTI and local data to create a coherent intelligence network. This unified representation is essential for enabling meaningful detections across diverse scenarios and sources.
- **Correlation Beyond IOC Matching:** The system must be capable of correlating structured SDOs with local intelligence in a way that supports detection of previously unseen scenarios. By leveraging the various aspects of the STIX framework, including relationships, attack patterns, and threat-actor context, ICARUS aims to identify malicious activity even when explicit IOCs or signatures are not directly present within the monitored environment.
- **Risk-Orientated Assessment of Correlated Events:** ICARUS requires a risk-scoring mechanism capable of evaluating the severity of the various correlated events while taking into account factors such as temporal decay, threat criticality, and contextual depth. This enables the system to elevate the ones estimated as presenting the most critical to the monitored environment, highlighting the most relevant threats.
- **Context-Rich and Actionable Alerting:** Alerts must provide a clear explanation of the reasoning vector behind each detection, including the entities involved and their relationships to both the threat and affected assets. This ensures that analysts not only receive a notification of potential compromise, but also the context needed to understand and respond effectively to possible not before seen incidents.
- **Autonomous Adaptability to Emerging Threats:** The system must continuously adapt to the ever-changing cybersecurity landscape by continuously incorporating newly published intelligence and adjusting its own data-collection mechanisms accordingly. This autonomy is the core of the ability of ICARUS to recognise unseen threats without requiring manual intervention or updates to pre-defined rule sets.
- **Modularity, Extensibility, and Scalability:** The architecture must remain modular and scalable, allowing ICARUS to integrate new intelligence sources, handle high volumes of data throughput, or expand its risk assessment capabilities as needed. This ensures long-term viability while supporting deployment in operational settings where load and data complexity can vary significantly.

Together, these goals form a cohesive foundation for the design of CTI-driven detection pipelines that are operationally robust and flexible to evolving requirements of organisations and environments. They ensure that the system is not only capable of gathering intelligence, but also of reporting accordingly and timely.

The following section builds on these principles and presents a modular system architecture that meets these requirements.

4.2 Architecture Overview

Following the design goals outlined in Section 4.1, the architecture of the proposed system, ICARUS, follows a modular and extensible design that facilitates the operationalisation of CTI. Its structure is intended to support the ingestion of data from local and external sources, aggregating and correlating this information within a main server that orchestrates the detection and alerting pipeline.

The main components are thus defined as follows:

- **CTI Providers**: These sources emulate external CTI feeds, supplying structured STIX data through TAXII-like interfaces. Their inclusion enables controlled and repeatable testing of the CTI ingestion mechanisms of the system. By simulating dynamic external feeds, this component validates the system's ability to continuously update its intelligence base, adapting to new threats as they emerge.

- **Endpoint Agents**: Deployed on monitored hosts, the agents collect local observables such as running processes and active network connections. Their role is to collect local intelligence and provide an environmental context that can be correlated with external intelligence. The deployment of these agents, rather than collecting data from various logs or monitoring systems, unifies the data collection process, ensuring consistency and reliability in the local data ingested by the main server.
- **Main Server**: Acting as the core of the system, the main server acts as the orchestration layer for ingestion, correlation, risk assessment, alert generation, and contextualisation, exposing a user interface for analyst interaction. It takes the role of the autonomous capabilities of the system, coordinating the full detection pipeline while remaining modular and reproducible. The server represents the central proof of feasibility for CTI-driven detection, directly addressing all three research hypotheses through its comprehensive functionalities.

Together, these elements form a complete solution capable of autonomous CTI-driven alerting. In Figure 2, an overview of the architecture of the system is presented, illustrating the main aspects that make up ICARUS and their interactions.

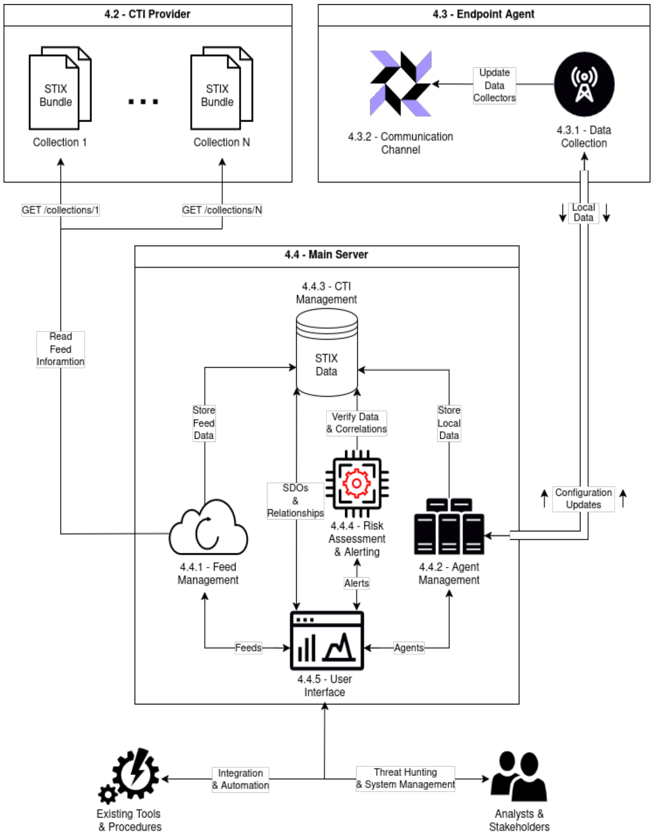

Figure 2: Overview of ICARUS' architecture, illustrating the main components and their interactions.

These components establish the foundation for an intelligence-driven detection pipeline in which external threat data and local intelligence are continuously correlated to generate contextualised alerts. The architecture therefore provides the structural basis for ICARUS.

The following sections describe the main functionalities of each component, focussing on the components that directly enable the operationalisation of CTI.

4.3 Detection Pipeline

In support of the system architecture presented in Section 4.2, the detection pipeline now takes the various aspects that make up ICARUS and defines the sequence of steps through which external CTI and local data are combined to identify, contextualise and alert for potential security incidents affecting the monitored environment. Figure 3 provides an overview of the complete detection pipeline, illustrating the progression from data ingestion to incident response.

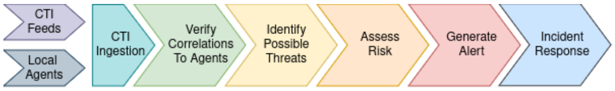

Figure 3: Overview of the system's detection pipeline.

As depicted in Figure 3, the main flow stars with intelligence ingestion, once again highlighting it as the foundational step for the entire process. The subsequent stages then build upon these data within the main server, progressively refining and contextualising the information until actionable alerts are generated.

In the following subsections, each stage of the pipeline is described, outlining its purpose and how it contributes to the overall goal of effective threat detection based on CTI.

4.3.1 *CTI Feeds and Local Agents.*

The detection pipeline begins with the acquisition of two complementary intelligence sources:

- **External Providers:** The main source of intelligence on emerging threats, attack techniques, and adversary behaviour. These sources supply structured data that encapsulates both indicators and relationships that can be used for both detection and contextualisation. By continuously updating the intelligence, ICARUS can adapt to new threats as they emerge, updating its detection techniques accordingly.
- **Local Agents:** Simultaneously, various endpoint agents monitor the local environment, collecting data about running processes, network connections, and other relevant system activities. This local telemetry provides the real-time context and history of the monitored hosts. These data points can then be correlated with external intelligence to identify, contextualise and alert for potential suspicious behaviour.

All collected data is thus forwarded to the main server, where it is processed through the subsequent stages of the detection pipeline.

4.3.2 *CTI Ingestion.*

Upon receiving data from both external providers and local agents, the main server initiates the ingestion process. This stage involves parsing the incoming intelligence, ensuring that both sources are normalised into a unified internal representation.

This includes validating the difference sources and creating indexes that facilitate efficient querying and correlation in later stages. While for external CTI this involves parsing STIX bundles and extracting relevant SDOs and relationships (along with their reported risk scores), for local data this process is expanded to directly create a connection between the collected observables and the originating endpoint agent. This linkage is essential for later stages, as it allows the system to trace back any gathered intelligence to its source, providing not only the basis for detection, but also context about the affected assets, when data was gathered, and other relevant metadata.

4.3.3 Verify Correlations to Agents. Following ingestion, the system then evaluates whether the newly acquired intelligence correlates monitored endpoints to known threats. This can either be through direct matches between indicators and local observations or through more complex relationships that link benign local activity to potentially malicious behaviour, such as command-and-control patterns or Advanced Persistent Threats (APTs).

For this to be possible, ICARUS leverages the relationships defined within the STIX framework to infer possible attack vectors between agents and threat entities. This step ensures that new data added to the system is continuously evaluated against the existing intelligence base, allowing for the timely identification of potential threats without the need for manual rule updates or signature definitions.

4.3.4 Identify Possible Threats. Once correlations are established, the next step involves identifying which of these correlations represent potential threats to the monitored environment. This acts as the first filtering layer, distinguishing between benign correlations and those that warrant further investigation.

During this stage, the system analyses the following aspects:

- **Base Threat Risk:** The inherent risk associated with the correlated threat entity is the first factor considered. Since it can vary significantly between different types of threats, this initial assessment helps prioritise only threats that are above a predetermined risk threshold for further analysis.
- **Previous Detections:** The vector connecting the local agent to the threat entity is also evaluated against historical detections. Since the same threat can present itself through various techniques or indicators, this step ensures that repeated detections of the same exact chain of SDOs are not redundantly escalated. This reduces alert fatigue and focusses attention on new or evolving threat patterns.

Vectors that pass these criteria are then escalated to the risk assessment stage for further evaluation of their potential impact on current infrastructure.

4.3.5 Assess Risk. In this stage, ICARUS is responsible for the comprehensive assessment of the risk posed by each identified threat vector. This assessment takes into account multiple factors to derive a holistic risk score that reflects the potential impact on the monitored environment. The factors considered include:

- **Threat Criticality:** The inherent severity of the threat entity, as originally reported by the external provider. This forms the baseline for the risk calculation.
- **Temporal Decay:** Building upon the base value and recognising threat evolution over time, the system incorporates a temporal decay factor into its risk estimation process. This is a valuable addition as it ensures that newer threats are prioritised.
- **Contextual Depth:** Finally, the depth of contextual relationships linking the threat to the local agent is evaluated. Threats that are directly correlated to a local agent see their risk score amplified, due to the increased likelihood of impact. However, as the number of intermediary relationships increases, the risk score is altered accordingly to reflect both the uncertainty and potential dilution of the threat's immediacy.

The combination of these factors results in a dynamic risk score for each threat vector, allowing the system to prioritise alerts based on their potential severity and relevance to the monitored environment.

4.3.6 Generate Alert. If the risk score estimated in the previous step exceeds the defined threshold, the next logical step is to generate an alert for the detected threat. This alert encapsulates all relevant information about the alert such as the risk score, affected agent, and date of detection. In addition, it includes detailed information on the entities involved, their relationships, historical context, and references to the SDOs that compose the detection vector.

4.3.7 Incident Response. The final stage is the incident response, where the generated alerts are presented to security the analysts through a dedicated user interface. This allows for the prioritisation and investigation of potential security incidents based on the risk scores and contextual information provided.

It contextualises the alert by presenting the threat vector both visually and descriptively, enabling both rapid comprehension and in-depth analysis. Following this, each SDO involved in the detection can be further explored, allowing analysts to understand surrounding details such as ongoing endpoint activity, related threats, and historical detections.

In summary, this pipeline operationalises cyber threat intelligence by continuously aligning external knowledge with endpoint observations. By automating the correlation, risk assessment and alerting processes, ICARUS transforms raw information into actionable insights, enabling proactive detection and response to the emerging threats.

5 Proof of Concept

Based on the architecture and pipeline described in Section 4, a proof of concept implementation was developed. This section details the main features of ICARUS, describing their design, functionality, and integration into the overall detection framework.

5.1 External Intelligence Ingestion

The first component of the system is responsible for the ingestion of external threat intelligence in a structured and repeatable manner. Its design mirrors common practices in contemporary CTI sharing ecosystems. The external provider is implemented to replicate the behaviour of operational CTI services in a controlled manner, allowing for consistent testing and validation of the ICARUS system.

The CTI provider follows the principles of TAXII collections, offering a clear request–response model for retrieving intelligence bundles. This approach effectively replicates the behaviour of CTI transmission services by allowing stakeholders to poll for new or updated intelligence.

Each collection is represented as a sequence of STIX bundles. A bundle groups together multiple STIX objects alongside their relationships, enabling a cohesive representation of threat entities and their contextual links. In this system, each bundle includes:

- **”type”** – Identifies the object as an STIX bundle.
- **”id”** – A unique identifier assigned to the bundle.
- **”objects”** – An array of STIX objects, such as indicators, malware, infrastructure, or network artefacts.
- **”relationships”** – A set of relationship objects describing how the STIX entities connect and interact.

This structure facilitates the ingestion of diverse intelligence, combining atomic indicators with contextual relationships that enable more meaningful correlation during analysis. The inclusion of information relating different objects is particularly important, as it reflects how adversaries behave and allows the internal components to infer links between objects detected in the system to possible threats rather than solely relying on the detection of isolated data points.

This component is configured with a list of collections stored as JSON files. To emulate dynamic intelligence streams, the available bundles can then be exposed incrementally or updated over time, simulating continuous evolution in the threat landscape, a characteristic feature of real-world CTI feeds.

This setup enables controlled experimentation while maintaining fidelity to operational CTI-sharing practices. By providing CTI that is structured, contextualised, and dynamically updated, the external provider forms a foundational element of the system’s ability to perform automated correlation and alert generation for previously unseen threats.

5.2 Local Data Collection

The endpoint agent is responsible for collecting local telemetry from the monitored hosts and delivering it to the main server for analysis. Its role is to continuously collect relevant system information that can be used for detection and correlation with external threat intelligence.

The agent works as a lightweight process designed to seamlessly integrate into the wider detection pipeline. It periodically collects data from the endpoint (e.g., running processes, network connections, and existing file hashes) and transmits this information to the main server. However, in order to ensure adaptability, the agent supports remote management from the server, allowing it to receive updated collection queries as needed, enabling dynamic adjustment of the data gathering strategy based on emerging threats or analytical requirements.

For the data collection process, the endpoint agente leverages *osquery*’s capabilities for extracting structured information from the host system. This decision is motivated by *osquery*’s flexibility, extensibility, and capability to gather information across different operating systems using a unified SQL-based interface that allows for direct mapping of collected data to STIX Domain Objects.

An example is shown in Listing 1, which retrieves metadata and cryptographic hashes for executable files in the folder ”/tmp”, a location frequently used for temporary or malicious artefacts.

```
1  SELECT
2      f.path, f.size, f.atime, f.ctime, f.mtime
           ,
3      h.md5, h.sha1, h.sha256
4  FROM
5      file f JOIN hash h
6      ON f.path = h.path
7  WHERE
8      (f.mode & 73) != 0
9      AND
10     f.directory IN ('/tmp');
```

Listing 1: Example *osquery* SQL query to retrieve executable files in “/tmp“ with their metadata and hashes.

In this example, both the file and hash tables are used to gather the necessary information. The file table records key properties such as its *path, size, access, creation, and modification timestamps*, while the hash table calculates cryptographic digests (MD5, SHA1, SHA256) for each file, allowing unambiguous identification across systems and feeds. The JOIN clause (Listing 1, line 5) is thus used to aggregate these complementary views, creating a single enriched record that captures both structural and integrity information for every file. This information is crucial not only for identifying potentially malicious files, but also for contextualising their presence and activity on the monitored host.

Queries like the one shown are then executed at regular intervals, with the collected results formatted as JSON objects that represent the relevant SDOs. These objects may be collected repeatedly over time or may represent large data dumps depending on the nature of the query. This makes the agent highly adaptable to different monitoring needs, demanding, on the other hand, a robust management and communication strategy to ensure the system’s efficiency and responsiveness.

5.3 Server-Agent Communication

This section details the establishment of a reliable communication channel. This component of the endpoint agent enables reliable and secure interaction with the main server, having two main functions:

- **Data Transmission** - The channel is responsible for transmitting the collected system telemetry to the main server in a structured and timely manner.

- **Remote Management** - The channel also facilitates the reception of new collector queries and configuration updates, allowing for dynamic management of the agent's data collection processes.

This bidirectional capability ensures the continuous delivery of host data while maintaining the flexibility to receive instruction updates in response to emerging threats. In addition, it sets the foundation for dynamic data collection, allowing the server to autonomously enable, disable, or adjust specific collectors based on the evolving threat landscape.

The server-client channel is implemented through a custom protocol layered on top of the base TLS, providing both confidentiality and integrity of transmitted data. This decision balances performance with robustness, defining an essential set of message types while remaining extensible for future enhancements. The protocol defines the following four primary message types:

- **"data"**(data) - This message type is used by the client to send collected data from the host to the server. Encapsulates the results of executed queries, allowing for easy parsing and direct conversion to SDO structures.
- **"upd"**(update) - This message type is sent by the server to update the client's configuration. It contains a set of new or modified queries that the agent should execute, allowing dynamic adjustment of data collection based on current needs.
- **"err"**(error) - This message type is used to report errors in the overall pipeline. The client or server can send it to indicate issues such as malformed messages, errors in data collection, or other anomalies that may require attention. This ensures that both ends of the communication can maintain robustness and reliability by providing feedback on the operational status of each other.
- **"ack"**(acknowledgment) - This message type serves as an acknowledgement for received messages. Ensures reliable delivery by confirming that a message has been successfully received and processed.

Each message is encoded as a JSON object that, while not the most efficient format in terms of size, is widely supported and easily integrated with various programming languages and frameworks. Thus, by adopting it, the system ensures that the results are both human-readable and easily processed by the server, facilitating debugging and future development.

From the client side, two dedicated loops manage communications by balancing incoming and outgoing messages. The sender loop periodically executes all the active queries, encodes their results as a "data" message, and transmits them to the server at the preconfigured interval. This ensures that the system continuously receives the relevant data without the need for a manual trigger, directly supporting the autonomy required by the hypothesis. On the other hand, the listener loop is reactive, waiting for messages from the server. For example, when an "upd" message is received, the agent dynamically updates its query set without interruption to its operation, demonstrating its adaptability.

These mechanisms are the foundation for the system's ability to adapt its collection practices dynamically. Its communication channel ensures that data is reliably transmitted while also allowing remote management and updates.

5.4 Data Storage and Correlation

This main server's data storage serves as the unified repository for all threat intelligence, be it from external feeds or local agents, providing the foundation for correlation, detection, and contextualisation. It is stored using structured STIX objects and relationships, ensuring its consistency across sources.

Before insertion, incoming data undergoes parsing and normalisation to ensure uniform a structure and the deduplication of entities, as different sources may report overlapping intelligence. To address this, ICARUS applies a canonical fingerprinting mechanism where non-essential metadata is removed (for the hashing process), while the remaining semantic content is serialised into a sorted JSON representation. The result is then hashed using SHA-256 obtaining a unique fingerprint identifier for the object's meaning rather than its original representation. This allows the system to merge duplicates by simply looking up existing fingerprints, without the need to exhaustively compare every field and value of the objects already stored, maintaining a coherent intelligence database.

Each stored object is complemented with essential metadata, including its identifier, type, sensitivity marking (according to the Traffic Light Protocol), initially assigned risk score, origin, and a modification history. These fields support traceability, enrichment, and filtering, enabling the system to maintain clear contextual awareness as objects evolve over time.

Beyond the individual SDOs, the storage keeps track of all relationships between objects, including the local agents, information later used to identify possible intrusion vectors. With relationships reported by external intelligence being stored alongside those derived from local data, the system can construct a comprehensive graph of interconnected entities. This intelligence network not only captures direct matches that may indicate compromise, but also expands the detection scope by revealing indirect associations and patterns. Furthermore, by linking local endpoint data to external CTI, it enables rich contextualisation potential during alert generation.

This approach enables the system to discover intrusion vectors that span multiple entities, such as, for example, a host reporting a benign-looking network connection that ultimately links to a known malicious infrastructure. These relational paths form the input to the risk assessment layer, where their severity is evaluated.

5.5 Risk Assessment Strategy

Once correlations between endpoint and possible threats are established, the system has to evaluate their importance through a dedicated risk assessment stage to identify relevant threats while maintaining an analyst focus on the most critical issues.

Each externally reported SDO contains a base risk score that reflects its inherent severity. However, to effectively estimate the actual risk, several factors need to be considered:

- **Source of Intelligence:** Different CTI providers may have varying levels of reliability and accuracy. The system incorporates base trust scores for each source that act as multipliers to the reported risk values, ensuring that intelligence from more reputable sources has a greater influence on the final risk assessment.

- **Correlation depth:** Direct correlations, i.e. when an endpoint directly reports an observable matching a threat indicator, are weighted more strongly than indirect ones (e.g., a benign process linked to a domain associated with a threat actor).
- **Temporal decay:** Risk scores are gradually reduced over time to prevent stale indicators from keeping disproportionate influence, ensuring that new intelligence is prioritised. However, in the case of, for example, Advanced Persistent Threats (APTs), where long-term campaigns are common, the external feeds can reintroduce previously known threats with updated information, effectively resetting (or even increasing) their original score.

These mechanisms ensure that the risk estimation process remains adaptive, taking into account both freshness, relevance, and source credibility.

However, in order to even begin the risk computation process, ICARUS first needs to identify all possible intrusion vectors that connect the monitored endpoints to known threat intelligence objects.

For this purpose, each monitored endpoint is represented as a graph linking local SDOs to related external intelligence objects, the central point being the agent itself. The traversal of this graph then identifies all paths that can be constructed from the agent node to any threat, henceforth referred to as *threat vector*. The system then filters out all paths that do not lead to possible malicious SDOs (i.e., objects containing zero risk scores).

For each remaining path, p, the risk is calculated taking into account several factors. The first is the base risk score of the threat object being evaluated, R. This score is then adjusted based on the depth of the path, d, not accounting for the agent node itself or the edges. The adjustment is done using a multiplicative factor, f, which is a configurable parameter that defines how much the risk should increase or decrease based on the path length, having it's base risk value be the same when the number of nodes (without the agent) equals f. Fitting the risk on a 0 to 100 scale, nodes that are closer to the agent have a higher impact on the overall risk (limited to 100), while those further away contribute less. Bellow, a comprehensive deduction of the risk computation formula is presented:

(1) The path p alternates between nodes and relationships, starting with the agent node. Therefore its total length is:

$$L = |p| = 1 + 2d$$

Where d is the number of nodes excluding the agent, i.e., the depth of the path. From this relation it follows that:

$$d = \frac{L-1}{2}$$

(2) The threat object has then a base risk score R. To adjust this risk according to path depth, a multiplicative factor f (the depth multiplier) is applied. The adjusted value is initially expressed as:

$$R \cdot f$$

(3) Since the risk value must remain within the defined interval, the system needs to limit the maximum score to the highest threat level (100):

$$\text{vector\_risk} = \min(R \cdot f,\ 100)$$

(4) To incorporate the path depth d, the adjusted risk is divided by d, which ensures that objects further away from the agent contribute proportionally less to the overall score. It also ensures that, due to the multiplicative factor, the risk remains the same when the path length (without the agent) equals this value i.e., when $d = f$:

$$\text{vector\_risk} = \min\left(\frac{R \cdot f}{d},\ 100\right)$$

(5) Finally, substituting $d = \frac{L-1}{2}$ back into the expression results in the final version for the risk assessment formula used in the POC:

$$\text{vector\_risk} = \min\left(\frac{R \cdot f \cdot 2}{L-1},\ 100\right)$$

This approach balances the factors mentioned above, creating a nuanced risk assessment that is deterministic but adaptable to the dynamic nature of cyber threats.

5.6 Alerting and Contextualisation

The final aspect of the system is the alerting component, which transforms the correlation and risk assessment results into actionable insights for security analysts. Based on identified potential intrusion vectors and their associated risk scores, the alert component now transforms the existing information into structured alerts that can be used directly in operational workflows.

Whenever a threat vector's computed risk exceeds a predefined threshold, the system generates an alert. This alert signifies that the relationship between a monitored host and a known threat presents a significant risk that warrants the attention of the analyst. Each alert includes the affected host, the trigger object, its calculated risk value, and the date of detection. Furthermore, in order to facilitate comprehensive analysis and better contextualise the alert, the system enriched the generated alert with a direct representation of the attack vector, as well as detailed information about all the SDOs and relationships that led to the detection.

New alerts are stored in an active collection that supports status changes, such as confirmation or dismissal. This lifecycle enables traceability and retrospective evaluation of detection performance, providing feedback that can be used to inform future parameter tuning in the risk estimation strategy or threshold settings.

To support broader situational awareness, the alerting module also monitors overall trends in stored intelligence risk values. It periodically calculates the mean risk for each object type (e.g., IPV4, Network Traffic, etc.) and uses it as feedback for adapting data collection strategies. When a particular value crosses a configurable threshold for that type of SDO, relevant collectors are activated (or deactivated) across the monitored hosts. This creates a lightweight mechanism that adjusts data collection efforts in response to observed shifts in threat activity without overloading the communication channel or the endpoints detection capabilities, ensuring continued relevance while limiting unnecessary overhead.

By combining structured information and vast context, ICARUS can generate actionable and informative alerts. Through autonomous adaptation, the system bridges the gap between simple data collection and detection of complex/unseen threats, adaptively focussing resources on the most pertinent aspects of the threat landscape. It provides analysts with a clear overview for each alert, ensuring that the system remains transparent in its operations and decisions.

6 Implementation Summary

The implementation of ICARUS brings together the main components of the system in a single, fully operational, pipeline. Each module is extensible, allowing the system to adapt naturally as new techniques and technologies emerge in the field of CTI and cybersecurity. The proof of concept is also designed to be replicable, enabling other researchers and practitioners to validate and build upon the presented work.

The complete dissertation that led to this paper, the full codebase, configuration files, and examples used in this work, are publicly available in the project repository:

- https://github.com/RicAlvesO/ICARUS

7 System Evaluation

- Intro to chapter

7.1 Methodology

- What data was used for evaluation
- What systems was it compared against
- What metrics were used
- Overview of possible limitations of the evaluation

7.2 Comparison of Results

- Base results for ICARUS
- Comparison against other systems

7.3 Analysis and Discussion

- Analysis of results
- Identification of strengths and weaknesses
- Discussion of implications
- Future improvements based on evaluation

8 Conclusions and Future Work

- Intro to chapter

8.1 Summary of Contributions

- Operationalisation of CTI
- New approach for detection of unseen threats
- Working and replicable open source implementation

8.2 Future Work

- Integration of advanced machine learning techniques for anomaly detection and threat contextualization
- Expansion of CTI sources
- Integration with SOAR platforms
- Critical assets prioritization

References

[1] Sean Barnum. 2012. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation* 11 (2012), 1–22.

[2] Sean Barnum, Robert Martin, Bryan Worrell, and Ivan Kirillov. 2012. The cybox language specification. *The MITRE Corporation* (2012).

[3] Sheng-Shan Chen, Ren-Hung Hwang, Asad Ali, Ying-Dar Lin, Yu-Chih Wei, and Tun-Wen Pai. 2024. Improving quality of indicators of compromise using STIX graphs. *Computers & Security* 144 (2024), 103972.

[4] David Chismon and Martyn Ruks. 2015. Threat intelligence: Collecting, analysing, evaluating. *MWR InfoSecurity Ltd* 3, 2 (2015), 36–42.

[5] Julie Connolly, Mark Davidson, and Charles Schmidt. 2014. The trusted automated exchange of indicator information (taxii). *The MITRE Corporation* (2014), 1–20.

[6] CVE.ICU. 2025. CVE Publications by Year. https://cve.icu/index.html

[7] cybermindr. 2025. The Race Against Exploitation: Average Time-to-Exploit in 2025. https://www.cybermindr.com/blog/average-time-to-exploit-in-2025/ Accessed: 2025-10-25.

[8] Anissa Frini and Anne-Claire Bourey-Brisset. 2011. *An intelligence process model based on a collaborative approach.* Defence R & D Canada.

[9] Yang Guo. 2023. A review of Machine Learning-based zero-day attack detection: Challenges and future directions. *Computer communications* 198 (2023), 175–185.

[10] Stefan Hagen. 2017. CSAF Common Vulnerability Reporting Framework (CVRF) Version 1.2. OASIS Committee Specification 01. http://docs.oasis-open.org/csaf/csaf-cvrf/v1.2/cs01/csaf-cvrf-v1.2-cs01.html Edited by Stefan Hagen.

[11] Ying He, Ellis Inglut, and Cunjin Luo. 2022. Malware incident response (IR) informed by cyber threat intelligence (CTI). *Science China. Information Sciences* 65, 7 (2022), 179105.

[12] Zafar Iqbal, Zahid Anwar, and Rafia Mumtaz. 2018. STIXGEN-a novel framework for automatic generation of structured cyber threat information. In *2018 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 241–246.

[13] Bret Jordan, Rich Piazza, and Trey Darley. 2021. STIX Version 2.1. OASIS Standard. https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html Standards Track Work Product. Copyright © OASIS Open 2022. All Rights Reserved..

[14] Panos Kampanakis. 2014. Security automation and threat information-sharing options. *IEEE Security & Privacy* 12, 5 (2014), 42–51.

[15] Inc. LevelBlue. 2025. AlienVault OTX. https://otx.alienvault.com/ Accessed: 2025-01-24.

[16] Francesco Marchiori, Mauro Conti, and Nino Vincenzo Verde. 2023. Stixnet: A novel and modular solution for extracting all stix objects in cti reports. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*. 1–11.

[17] Robert Martin, Steven Christey, David Baker, and MITRE Corporation. 2002. The Common Vulnerabilities and Exposures (CVE) Initiative. *MITRE Corporation* (2002).

[18] Andrew Ramsdale, Stavros Shiaeles, and Nicholas Kolokotronis. 2020. A comparative analysis of cyber-threat intelligence sources, formats and languages. *Electronics* 9, 5 (2020), 824.

[19] Langley Rock, Stefan Hagen, and Thomas Schmidt. 2024. Common Security Advisory Framework Version 2.0 Errata 01. OASIS Approved Errata. https://docs.oasis-open.org/csaf/csaf/v2.0/errata01/os/csaf-v2.0-errata01-os.html Edited by Langley Rock, Stefan Hagen, and Thomas Schmidt.

[20] Farhan Sadique, Sui Cheung, Iman Vakilinia, Shahriar Badsha, and Shamik Sengupta. 2018. Automated Structured Threat Information Expression (STIX) Document Generation with Privacy Preservation. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 847–853. https://doi.org/10.1109/UEMCON.2018.8796822

[21] Daniel Schlette, Fabian Böhm, Marco Caselli, and Günther Pernul. 2021. Measuring and visualizing cyber threat intelligence quality. *International Journal of Information Security* 20, 1 (Feb. 2021), 21–38. https://doi.org/10.1007/s10207-020-00490-y

[22] Daniel Schlette, Manfred Vielberth, and Günther Pernul. 2021. CTI-SOC2M2–The quest for mature, intelligence-driven security operations and incident response capabilities. *Computers & Security* 111 (2021), 102482.

[23] The MISP Project. 2025. MISP: Malware Information Sharing Platform. https://www.misp-project.org/ Accessed: 2025-01-24.

[24] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. 2019. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security* 87 (Nov. 2019), 101589. https://doi.org/10.1016/j.cose.2019.101589

[25] YesWeHack. 2025. CVE surge: Why the record rise in new vulnerabilities? https://www.yeswehack.com/news/cve-surge-record-jump-vulnerabilities