CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	KIDS: intrusion detection for industrial control systems
Authors	Wani, Nowshaba Jeelani;Pesch, Dirk;Roedig, Utz
Publication date	2025-08-09
Original Citation	Wani, N. J., Pesch, D. and Roedig, U. (2025) 'KIDS: intrusion detection for industrial control systems', in Coppens, B., Volckaert, B., Naessens, V. and De Sutter, B. (eds) Availability, Reliability and Security. ARES 2025. Lecture Notes in Computer Science, 15994, pp 191–208. Springer, Cham. https://doi.org/10.1007/978-3-032-00630-1_11
Type of publication	Conference item;book-chapter
Link to publisher's version	10.1007/978-3-032-00630-1_11
Rights	© 2025, the Author(s), under exclusive license to Springer Nature Switzerland AG. This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant number 18/CRT/6222. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-11-01 17:05:31
Item downloaded from	https://hdl.handle.net/10468/17854

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

KIDS: Intrusion Detection for Industrial Control Systems

Nowshaba Jeelani Wani, Dirk Pesch, and Utz Roedig

School of Computer Science and IT, University College Cork, Cork, Ireland
nowshabawani@umail.ucc.ie,{d.pesch,u.roedig}@cs.ucc.ie

Abstract. The convergence of Information Technology (IT) and Operational Technology (OT) has significantly increased the vulnerability of Industrial Control Systems (ICS). Prolonged undetected intrusions and the frequent exploitation of zero-day vulnerabilities have made ICS highly susceptible to cyberattacks, resulting in data loss and physical damage. Despite growing threats, majority of Intrusion Detection Systems (IDS) ignore the significance of process-based data and equipment such as Programmable Logic Controllers (PLCs) and focus on the management components of ICS, which are essentially an IT system. Many suggested IDS are also only effective with known attacks and fail to detect zero-day exploits. The lack of a unified IDS across IT and OT, applicable irrespective of protocols employed or hardware heterogeneity, is another significant gap in this field. This paper presents Kestrel-Based Intrusion Detection System (KIDS), a query-based, process-aware framework tailored for OT. Built on the Kestrel threat hunting language, KIDS combines process monitoring with traditional threat intelligence to detect sophisticated attacks across all layers of ICS. By abstracting system components and complexities into unified query interfaces, KIDS enables holistic visibility, from management systems to PLCs, and supports scalable, cross-platform threat detection adaptable to evolving industrial threats.

Keywords: Industrial control systems · Intrusion detection systems · Operational technologies · Threat hunting · Security.

1 Introduction

Industrial control systems are increasingly targeted by sophisticated cyber attacks, often resulting in operational disruptions, data breaches, or physical damage. Incidents such as Stuxnet [10], the Ukrainian power plant attack [14] have demonstrated how vulnerable critical infrastructure can be when targeted through both traditional and novel attack vectors. A significant contributor to this risk is the convergence of Information Technology (IT) and Operational Technology (OT), as industries integrate networked and remote capabilities into systems originally designed to be air-gaped and standalone.

The exposure of OT components such as PLCs, sensors and actuators, to external networks introduces a threat surface that conventional security tools,

particularly network based intrusion detection systems are ill equipped to handle. These IDS solutions, often implemented and manged by IT teams, typically focus on network traffic and management interfaces such as Human Machine Interface (HMI) and Data Historians. As a result, they largely overlook operational context and the behavior of underlying physical processes which are critical to understanding threats specific to ICS.

Moreover, traditional IDS models are heavily reliant on predefined rules or known attack signatures, limiting their effectiveness again zero-day threats, Advanced Persistent Threats (APTs) or attacks that subtly manipulate physical processes. Sophisticated adversaries can compromise ICS not just by exploiting software vulnerabilities but by introducing malicious control logic or altering operational thresholds, techniques that evade detection by systems blind to process dynamics.

To address these shortcomings, we propose KIDS, a Kestrel based IDS tailored for Industrial Internet of Things (IIoT). KIDS is a process aware detection framework that bridges the IT and OT divide by enabling security analysts to issue unified queries across network level and process level data. Built on the open source Kestrel [15] threat hunting language. KIDS introduces a flexible mechanism, to hunt for indicators of compromise across ICS layers, from Programmable Logic Controller (PLC) variables to network authentication logs. Additionally, once a hunt script is tuned for a particular scenario, it can be automated to run periodically as required, making KIDS a fully automated and customizable IDS to fit varying needs of unique ICS setups.

The key motivation for using Kestrel is that it already provides reusable and structured threat hunting queries for the IT domain and offers a high degree of flexibility. KIDS extends this capability by incorporating Kestrel in a high level IDS framework for threat hunting across IT and OT, using metadata about ICS processes (e.g critical variables, process IDs), enabling the corelation of behavioral anomalies with security events. This integration allows for proactive identification of stealthy threats that target control logic, manipulate sensor data or induce subtle deviations in physical behavior; attack strategies that are commonly invisible to the traditional IDS.

The appendix of this paper demonstrates how a typical kestrel script is adapted to hunt lateral movement attacks using Windows Management Instrumentation (WMI), showcasing the language's flexibility. This serves as a foundation for expanding the same approach to ICS environments through KIDS, where process integrity, control state deviations and contextual metadata become central to intrusion detection.

In summary, our work contributes a scalable, extensible and vendor neutral threat detection framework for ICS providing the following:

- An extension of kestrel for ICS threat hunting includes OT data sources such as PLCs and Historians.
- A unified query abstraction that enables correlation of IT and OT events in a single hunt.

- A metadata driven approach that contextualizes process variables and operational behavior.
- An evaluation using a testbed implementation of KIDS and MITRE ATT&CK framework for ICS, demonstrating KIDS ability to detect complex attack chains.

By introducing operational semantics into threat hunting , KIDS enhances visibility, reduces false positives and provides a unified tool for securing ICS environments.

2 Background

2.1 Examples of ICS Breaches

Recent years have witnessed several attacks on ICS, but only a few were sophisticated enough to cause any physical damage. Among them, Stuxnet [7] remains the most complex. It exploited multiple zero-day vulnerabilities to alter centrifuge operation logic on the PLCs. The payload included a PLC rootkit recorded previous input and output values, mask anomalous behavior, allowing the malware to evade detection and ultimately damage the centrifuges [10].

A separate attack on a German steel mill caused critical control failures. The safe shutdown sequence, including the blow-down process, may have been bypassed, resulting in an uncontrolled furnace state. While there were no injuries, the incident led to significant physical damage [13]. Another major incident occurred in Ukrainian (2016), where attackers remotely accessed the HMI to disable switch-gears on the grid, log out operators, change passwords, wipe disks, and overwrite firmware, rendering the hardware inoperable. This caused outages for roughly 230,000 inhabitants for 1–6 hours [14].

More recently, UK's Sellafield nuclear site was reportedly breached by sleeper malware, detected as early as 2015. The malware is suspected to have compromised activities such as handling radioactive waste, monitoring leaks without detection [1]. These incidents underline how attackers often infiltrate systems months before executing the payload, highlighting the need for timely detection to prevent damage.

2.2 Intrusion Detection and Threat Hunting

Intrusion detection involves monitoring networks and systems for signs of malicious activity, serving as a critical defense mechanism against cyber threats and enabled using intrusion detection systems. These help detect unauthorized access, policy violations, data breaches, and other types of threats. Depending on deployment and methodology, IDS can be host-based, network-based, or hybrid, and may use signature, anomaly, or behavior-based detection techniques. Despite their utility, IDS often struggle with high false positives, limited detection of zero-day threats, and rigidity. These limitations can be mitigated by incorporating proactive strategies such as threat hunting [17], [20]. It involves actively

searching for threats that evade conventional defense, using tools like Security Information and Event Management (SIEM), Event Detection and Response (EDR), threat intelligence, manual analysis. Integrating threat hunting into an IDS can enhance detection of stealthy threats, improve situational awareness, and strengthen overall security posture.

2.3 Threat Hunting with Kestrel

Threat hunting typically relies on tools such as Osquery, YARA, Zeek, Snort, and Wireshark, as well as languages such as SQL, YARA rule language and Powershell. However, threat hunters often face repetitive tasks such as creating scripts for varying Tactics, Techniques and Procedures (TTP), data sources, and environments, alongside handling complex data processing, handling threat intelligence, analytics setup, and hypothesis development. These efforts must often be repeated across diverse contexts, making the process tedious and inefficient. To address this, IBM introduced Kestrel [16], an open source threat hunting language. Kestrel enables reusable, composable hunt scripts and supports built-in and custom analytics (e.g., suspicious scores via Docker-based modules). It integrates with various EDRs, SIEMs, and other monitoring tools. Kestrel introduces entity-based reasoning, which essentially means building a hypothesis focused around entities (system, network, device) and what to hunt rather than how to hunt. It also supports forking and merging of hunts for ease of composability. An example Kestrel script for detecting lateral movement via Windows Management Infrastructure (WMI) on Windows systems is available online [4], demonstrating its utility for expressing complex threat hunting logic in reusable form.

3 Related Work

Numerous solutions have been developed to secure cyberspace such as IDSs. Khraisat et al. [12] review various methods and datasets for intrusion detection. They discuss the strengths and limitations of signature-based, anomaly-based, and behavior-based systems, along with evaluation techniques such as statistics-based methods. Several commonly used datasets for evaluating IDSs are also reviewed. Their work highlights general approaches in cyber intrusion detection but does not consider the unique needs of ICS environments, as proposed in our work.

Wang [20] reviews threat hunting literature and implementation methods, showing machine learning-based implementations to detect threats are the most popular. He also compares different threat hunting methods, such as intel-driven, observed data-driven, and TTP-driven approaches. Aldauiji et al. [3], review different threat-hunting models for detecting ransomware. They propose combining practical approaches with various models to detect unknown ransomware effectively. They also review the currently available ransomware datasets used in research. Both studies address proactive approaches but are focused on generic

IT systems, and do not integrate operational intricacy or process control into detection.

Authors Jadidi and Lu [11] propose a unified threat-hunting framework for IT and ICS. This Threat Hunting Framework (ICS-THF) targets early identification of threats. It consists of three stages: threat hunting triggers, threat hunting, and cyber threat intelligence. The trigger stage detects events or external cues to initiate hunting. The hunting stage combines the MITRE ATT&CK Matrix and Diamond model to form and validate hypothesis and predict adversary behavior. The final stage identifies Indicators of Compromise (IoC) for future use as threat intelligence. While this work is one of the few to bridge IT and ICS, it does not incorporate process-level metadata into the IDS framework to provide operational context, which is unique to KIDS.

In a study, Ajmal et al. [2], use threat hunting in conjunction with cyber deception and kill chain to detect attacks, including unknown adversaries. They introduce a decoy farm to lure attackers to a non-critical, deceptive setup within the ICS environment. Evaluation was conducted using several attacks on an emulated Supervisory Control and Data Acquisition (SCADA) network, compared with reactive approaches such as Zeek Bro, firewall (Palo Alto), and Windows Defender. The final results show that the threat-hunting approach is 60% faster and can detect known and unknown attacks. While the work emphasizes proactive detection and performance improvements, it does not address process-aware security or interpret ICS data in threat detection.

Bibi et al. [9] proposes a threat intelligence and detection method using convolutional neural networks Long Short Term Memory 2 Dimensional (LSTM2D). It is described as self-sufficient and self-optimizing, overcoming limitations of other models and dynamically detecting sophisticated threats in IIoT. They evaluated the model on a dataset with 21 million attack patterns and vector instances, showing high detection accuracy with a trade-off in speed. In contrast, our work does not rely on deep learning but leverages flexible scripting, annotated process metadata, offering interoperability and integration of control systems in the detection process.

Given the heterogeneous nature of ICS devices, securing diverse data is complex. Yazdinejad et al. [21] introduce an ensemble deep learning model with Long Short Term Memory (LSTM) and auto-encoder to identify unknown threats. The LSTM model learns the standard patterns of behavior while the autoencoder reduces data dimensionality. The model accounts for dataset imbalance using balanced subsets. Evaluated on Gas Pipeline (GP) and Secure Water Treatment (SWaT) datasets, it achieved 99.3% and 99.7% accuracy, respectively. Unlike our framework, this model does not enable correlation across systems, nor does it integrate control-specific annotations.

Arafune et al. [6], discuss an automated method of extracting threat intelligence, along with generating and validating hypotheses for threat hunting. The work uses MITRE's ATT&CK framework for threat intelligence. It also leverages open-source tools such as Elasticsearch, Conpot, Metasploit, Web Single Page Application and a machine learning analyzer. The paper highlights the need for

more research in automating threat hunting in ICS networks. Unlike other solutions that are cloud-based, expensive, and prone to human errors, this solution is central and open-source. The authors demonstrate that the solution can identify network attacks and alert a threat hunter with a hypothesis generated based on TTPs from ICS MITRE ATT&CK. They also employ a machine learning classifier to automatically predict future actions of the attack. This work emphasizes automation and hypothesis generation, whereas, KIDS focuses on context aware detections that can identify violations of physical process constraints.

Bhardwaj and Goundar [8] propose a framework for effectively implementing cybersecurity threat Hunting. The framework aims to empower organizations with the tools and methodologies to enhance their threat-hunting capabilities. Additionally, the proposed framework is compared against existing security operations using 20 key Threat Hunting metrics. Their approach, while methodical, focuses on aspects of threat hunting programs rather than ICS specific adaptations.

IIoT systems rely on many edge devices, including sensors, controllers, and robots, for various tasks such as data collection, transmission, storage, and processing. However, any malicious or abnormal behavior exhibited by these devices can compromise the security of the entire IIoT network. Additionally, such devices can serve as entry points for malicious software to infiltrate the network. To address these concerns, Yazdinejad et al. [22] proposes a parallel ensemble model for threat hunting that focuses on detecting anomalies in the behavior of IIoT edge devices. The model is designed to be flexible and can utilize multiple state-of-the-art classifiers as essential learners. It efficiently classifies multi-class anomalies using Multi-class AdaBoost and majority voting techniques. Experimental evaluations on a dataset comprising standard records from multiple sources and multi-class anomalies demonstrate that the proposed model outperforms existing approaches in accuracy, F1 score, recall, and precision. Our framework differs by focusing on control systems, operational context and integration of OT and IT data sources via Kestrel.

As evident from the works discussed in this section, the current scope of study in this field focuses on the management and network level of the ICS, completely neglecting the lower-level vulnerabilities, such as on the PLC level. There needs to be a standard solution that dives into the granular specific and can detect threats across the entire ICS setup. The approach proposed in this paper overcomes this gap and looks at the vulnerabilities throughout the ICS systems, including lower-levels.

4 KIDS: Kestrel-Based Intrusion Detection System

Classic IDS are traditionally deployed as dedicated network monitoring systems for traffic within an organization's IT infrastructure. They analyze network communications and raise alerts when they detect anomalies, either known attack signatures or deviations from the baseline behaviors. Such IDS solutions are typically operated by the IT departments, tasked with maintaining enterprise

security. This approach is effective in conventional IT environments, where attackers use a standard sets of exploits.

However, as discussed, this standard approach has its limitations when applied to a hybrid IT/OT setup. First, traditional IDS solutions offer little to no visibility into the OT domain. They primarily observe IT-layer components, such as servers, authentication systems and management interfaces, while remaining blind to the underlying industrial processes. This invisibility is reinforced by organizational silos, while IT manages the IDS, OT systems are designed and maintained by process engineers, with minimal coordination between the two groups. Second, industrial environments are inherently heterogeneous. They encompass a wide range of devices, vendors, and protocols, resulting in a far broader and more complex attack surface than typically seen in pure IT environments. Consequently, an effective IDS in this setting must be tailored to the specific operational context of the facility and the latest threat landscape. KIDS addresses the two aforementioned shortcomings for effectively implementing an IDS in a hybrid IT/OT.

Improved IDS Visibility: Kestrel provides abstractions and interfaces defined for IT environments, for manual threat hunting. It enables security analysts to gather security relevant information via queries across diverse systems using standardized abstractions. For example, analysts can issue requests to gather relevant data, such as "Were there any user logins between 2am and 5am?" or "Which systems executed a large number of processes between 2am and 5am?". These queries are modular, composable and reusable. Combining the two examples, we can now search for unusual timed logins together with unusual activity which may reveal an ongoing attack. We can also use these refined hunt scripts in an automated context and run it periodically, essentially forming an IDS. We extended the Kestrel framework to form KIDS to provide interfaces for OT infrastructure components. With that, you may now extend the example and include queries such as "Were any critical process variables outside their safe operating bounds?". Now, it is possible to see if there were unusual logins linked to unusual system activity and unusual behavior in the physical process. This of course requires an understanding of what critical variables are and what bounds may be. We envision that these contextual annotations are provided by process engineers designing automation. This approach provides a defined process to bridge the communication gap between OT engineers and IT staff safeguarding the infrastructure.

Improved IDS Customization: As outlined, it is possible with KIDS to compose hunt scripts and customize them to look for different security aspects in a combined IT/OT infrastructure. As production environments vary vastly it is necessary to adapt the IDS. For example, what is considered normal in an automotive factory might be considered anomalous in power plant. The Kestrel abstraction provides a scripting interface that abstracts from the underlying heterogeneous infrastructure. Adaptation to a use case is possible without interacting with individual OT components (once a KIDS abstraction is available for a component).

In the following sections, we provide more detail on KIDS and show how it can be used in specific contexts. We describe examples of specific KIDS/Kestrel syntax used to implement queries such as "Were there any user logins between 2am and 5am?". Figure 1 presents a high-level overview of the KIDS framework architecture and its core design elements.

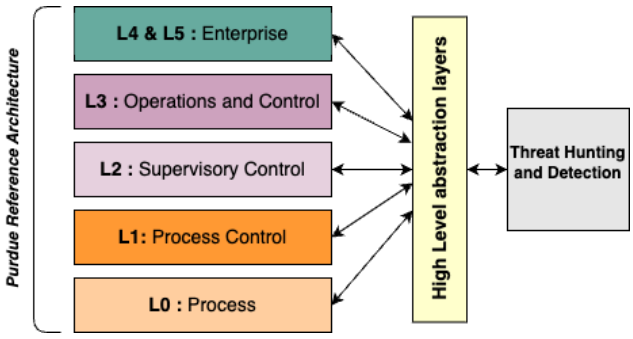


Fig. 1. KIDS, high-level view of the framework.

4.1 Testbed Environment

To demonstrate the KIDS implementation a small scale but comprehensive testbed was constructed, emulating a modern manufacturing factory. We also use this testbed to evaluate KIDS in a realistic setting. This testbed captures key architectural components typically present in industrial environments:

- **Control System:** The control logic is executed by OpenPLC [5], deployed on a raspberrypi 5. This simulated PLC drives a six servo-based robotic arm, simulating tasks like material handling or assembly.
- **Engineering workstations:** A windows Virtual Machine (VM) serves as the primary engineering interface for developing, compiling and uploading PLC programs via OpenPLC editor.
- **Human-Machine interface:** Two HMIs are used; OpenPLC built-in web interface and ScadaBR [18] on a windows VM is used for real time monitoring and manual control.
- **Data Historians:**
  - *SQLite:* serves as the Historian for process variables (e.g, servo motor angles, alarm status).
  - *MySQL:* Captures IT centric security events, such as SSH login attempts and system access logs.

All components communicate over a secure LAN, closely mirroring the communication patterns in actual factory floor systems. Figure 2 shows a reference ICS architecture [19] and Figure 3 our physical testbed implementation.

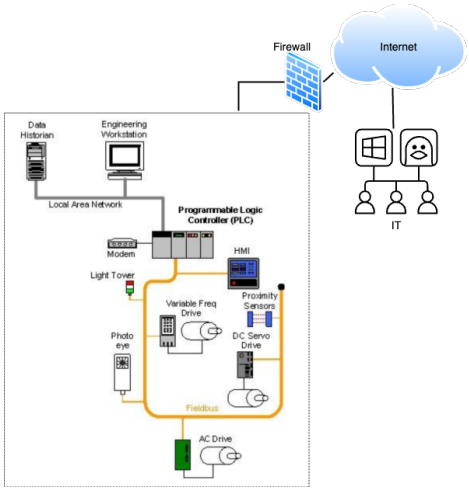


Fig. 2. Reference PLC control system from NIST 800-82.

4.2 KIDS Implementation in the Testbed

The core design philosophy of KIDS revolves around a unified query interface. It allows analysts to issue a single, abstracted command to retrieve data from various sources, whether it's firewall logs from SIEM or process variable readings from a PLC. This capability is built on top of the Kestrel threat hunting language, an open source tool that enables structured, reusable threat hunting scripts.

Key innovations in KIDS include: First, **process-aware querying** enables operational data such as servo motor angles or alarm states can be analyzed with the same ease as traditional log files. Second, **contextual annotation** enriches ICS process variables with metadata including threshold values, process IDs to aid interpretation, even for security analysts without domain expertise in ICS. Third, **cross-domain correlation** is made possible by linking events across IT and OT domains; for example, KIDS can relate a failed SSH login attempt to anomalous PLC behavior within the same analytical framework.

To address the challenge of heterogeneity in ICS, KIDS uses *custom data source interfaces* that standardize diverse inputs into a consistent schema (Structured Threat Information Expression (STIX)). These interfaces translate operational data and control logic annotations into structured threat intelligence, supporting rich correlation and automated detection across systems. The implementation of KIDS in the testbed involve several components working in unison to provide complete threat visibility. The design consists of a set of customizable modules that allow KIDS to access, interpret and correlate process-level data from industrial environments in the same way Kestrel language typically does with conventional IT data:

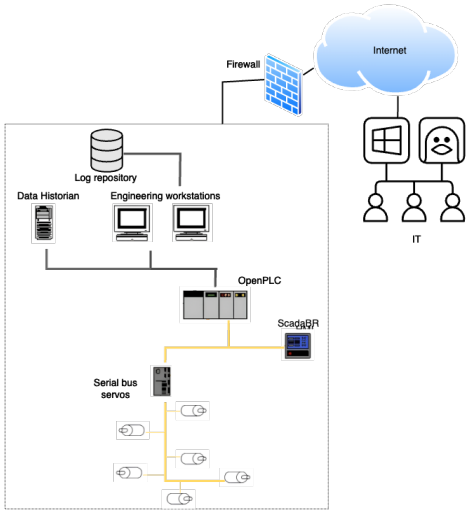


Fig. 3. Testbed implementation with Raspberry Pi and robotic arm.

1. Custom Data Source Interfaces: A fundamental concept that makes Kestrel structured, reusable and flexible are the Kestrel "interfaces", which connect the hunting logic to a data source. Interfaces in Kestrel decouple the "what" and "where" of the hunted data from "how" to query it. Each interface id registered with a unique URI-like schema prefix (e.g., stixshifter://, kestrel://, or custom interfaces like plc://) that tells Kestrel how to fetch, translate and transform data from the associated backend (such as EDRs, SIEM). These interfaces are enabled by underlying plug-ins that translate Kestrel commands (like GET) into backend native queries (e.g., MySQL), transform retrieved data into STIX objects and then return results into Kestrel's in-memory data store (firepit). The Kestrel language currently provides support for numerous standard interfaces such as:

STIX-Shifter Data Source Interface (stixshifter://) for numerous commercial and open-source EDRs, log repositories such as IBM QRadar, Elastic Search and so on.
STIX bundle Data Source Interface (file:///, https://, http://) for canned STIX bundle data.
Analytics interfaces such as Python Analytics Interface (python://) and Docker Analytics Interface (docker://).

To adapt Kestrel to ICS environments, we develop custom data source interfaces tailored for such systems:

plc://data: connects to the SQLite historian.
plc://openplc: retrieves runtime data from OpenPLC in near real-time.
plc://metadata: loads the contextual YAML annotations provided by the automation engineers.

stixshifter://mysql: pulls authorization and security events from MySQL.

These custom interfaces follow the same principles as standard Kestrel interfaces but are adapted to industrial systems and data formats. they enable Kestrel to ingest and normalize data from both OT and IT data sources into STIX format, facilitating correlation and analysis within Kestrel's in-memory environment.

2. Annotated Process Metadata: In ICS, security tools lack process context and domain semantics. This leads to blind spots, where even a valid Modbus command can produce dangerous states if not interpreted in relation to the underlying industrial processes. Annotated metadata bridges this gap, allowing Kestrel to understand what constitutes unsafe or unexpected process behavior. These Annotations serve as a structured medium through which automation engineers convey industrial semantics, such as which variables are critical, which processes are active on which PLC and what operational states and thresholds should not be violated. this improves detection accuracy and reduces false positives.

Automation engineers encode these annotations in a YAML file that defines but is not limited to:

- PLC IDs and their associated processes.
- Definitions of critical variables (e.g, angle, voltage, pressure, Revolutions per minute (RPM)).
- Safe operating thresholds.

This metadata is loaded into KIDS and enables process-aware detections. For example:

```
GET x-openplc-int FROM plc://plcdata WHERE name = 'angle'
```

can retrieve the current value of the process variable "angle". KIDS uses the annotated limits to flag anomalies where thresholds are exceeded, regardless of whether the triggering action appears legitimate.

3. Scripted Threat Hunting and Process-aware Correlation: Using KIDS, analysts construct modular threat hunts using Kestrel's primitives: "GET" for data retrieval, "JOIN" for correlation, and "DISP" for visualization. These operations allow complex hunts to be executed across multiple data layers. A typical hunt script may proceed as follows:

1. Query plc://plcdata for abnormal process variable values.
2. Retrieve login events from stixshifter://mysql.
3. Use "JOIN" to correlate anomalies in process telemetry with unauthorized user activity.

This approach enables detection of overlaps between suspicious IT actions and OT anomalies. Analysts can explore relationships between entities without requiring deep domain expertise in control systems. Figure 4 shows how Kestrel is

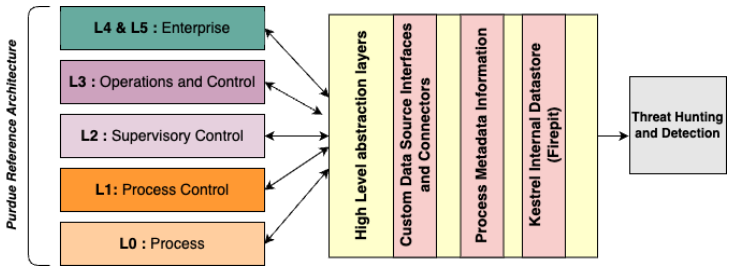


Fig. 4. KIDS implementation with Kestrel and custom data source interfaces.

integrated in KIDS framework to enable detection. Data from SQLite, OpenPLC, YAML annotations and MySQL are routed via interfaces into Kestrel's memory, where hunt logic processes them.

By combining process metadata with real-time telemetry, KIDS provides a rich, multilayered view into ICS security, empowering both IT and OT teams to detect and respond to threats collaboratively.

5 Evaluation

Evaluating an IDS for industrial environments poses unique challenges. Unlike IT security systems, where large scale benchmarks, public datasets and competitive baselines exists, evaluating a threat hunting based IDS, especially the ones targeting control layer anomalies requires carefully curated testbeds and contextual simulations. Furthermore, our proposed system, KIDS, is designed to focus on detecting nuanced, process level deviations that cannot be fully assessed using packet traces or standard SIEM logs alone. In this work, we evaluate KIDS through:

- **Controlled Testbed Attacks:** Simulating real-world ICS threats in our lab testbed and assessing KIDS detection capabilities.

- **MITRE ATT&CK for ICS:** Using MITRE ATT&CK navigator to benchmark which attack techniques can be recognized by KIDS in general.

This allows us to ground our results in both practical and theoretical frameworks. The evaluation focuses primarily on detecting *operational level attacks* (e.g, unauthorized process changes, sensor spoofing), as traditional network-based detection (e.g, lateral movement) is already well-handled by Kestrel's available connectors for various EDRs and SIEMs and log analytics platforms. Thus, we emphasize gaps left by traditional IDS and demonstrate how KIDS extends coverage into physical process domain strengthened further by the default capabilities of Kestrel.

5.1 Evaluation in the Testbed:

We base our evaluation on the testbed described in Section 4, where a raspberrypi controlled robotic arm performs repetitive tasks simulating a manufacturing/assembly line. Critical variables such as servo motor angles and LED alarms are annotated by the automation engineer using YAML metadata file. These annotations expose what the process engineer considers safety-critical or indicative of abnormal process behavior. KIDS interfaces with the Historian (SQLite), OpenPLC runtime and metadata files using custom interfaces (see Section 4). A baseline KIDS hunt script is constructed, querying and joining OT and IT records. Once tuned manually, these scripts can be automated to run periodically as an automated IDS. We emulated threats such as:

1. **Unauthorized Modbus commands to change process setpoints:** The attacker compromises a remote engineering workstation and sends a legitimate looking Modbus read and write commands directly to the robotic arm, altering the target angle to an unsafe value (e.g, 140 degrees when the safe limit is 120). Traditional IDS tools observe no anomaly because the packets are valid and authenticated. KIDS, however, correlates the timing of this Modbus access with a recent login and observes the arm's angle exceeding its annotated safety threshold.
   The hunt script flags this as a potential process violation. Kestrel's "GET" retrieves both login and PLC variables, and "JOIN" correlates the two based on timestamps. The use of process-aware context enabled by annotated process metadata makes its evident that although the access was technically legitimate, the result was an unsafe operational state.

2. **Injection of new PLC logic via valid but unauthorized users:** An attacker uploads new operating logic to the PLC via OpenPLC's web interface using a stolen engineer credential. The uploaded program includes a subroutine that overrides safe limits under certain conditions. KIDS detects this configuration drift by querying the process metadata and comparing it to the new events. This inconsistency would go unnoticed in traditional tools. Instead of using complex and explicit process signatures or rules written for each logic variation, KIDS simply highlights this deviation from expected process variable behavior.

3. **Suppression of alarm triggered by crossing the operational thresholds:** The attacker modifies the LED alarm output variable to remain inactive, even when the critical thresholds are crossed. During testing, angle values breach thresholds, but the alarm is not triggered. Using historical trends from SQLite Historian and variable thresholds from process metadata, KIDS identifies abnormal suppression of the LED alarms system in response to out-of-bounds values. This form of suppression can also mimic hardware fault rather than an attack, which can be dangerous. KIDS correlation reveals contradiction between expected and observed process behavior.

The above mentioned attack strategies mimic techniques employed in sophisticated ICS attacks like *Stuxnet*, *Industroyer* and *PLC-Blaster*. They highlight KIDS's unique strength in process-aware security monitoring and detecting subtle manipulations that blend in at the protocol level but disrupt operations.

5.2 Evaluation Using the MITRE ATT&CK for ICS:

The MITRE ATT&CK framework for ICS is a structured knowledge base of known adversarial behaviors. It catalogs tactics and techniques that adversaries use across the kill chain, from initial access to impact, within ICS contexts. By evaluating KIDS against MITRE ATT&CK for ICS, we aim to assess how well our proposed system covers known tactics used in real-world intrusions. Since Kestrel already provides robust integration with many IT security tools, our evaluation is centered on process level impact and operational disruptions; areas often invisible to security solutions.

To perform this evaluation, we first simulated representative techniques of attacks such as Stuxnet, Industroyer and PLC-Blaster within our testbed, as described in the section 5.1. Then using the ATT&CK Navigator tool, we mapped:

1. The techniques employed during these simulated attacks, colored in red (Industroyer), yellow (Stuxnet), blue (PLC-Blaster) and orange (techniques used two or more attacks)).
2. The techniques KIDS was able to detect (highlighted in green).

Figure 5 presents a view of the techniques employed by major ICS malware campaigns such as Stuxnet (e.g., T0832 - Manipulation of Control, T0812 - Logic Injection), Industroyer (e.g., T0810 - Alarm Suppression, T0855 - System Firmware) and PLC-Blaster (e.g., T0803 - Unauthorized Change). These techniques are color-coded for clarity as described above. This visualization reflects the diversity of tactics employed across phases such as Execution, Inhibit Response Function and Impact. It also highlights the complexity of multi-phase attack chains used to compromise both IT and OT.

Figure 6 provides a view of the techniques described above, that might be detected by classic IDS such as Snort. This alongside the figure below, shows that many techniques on the operational side are undetectable. KIDS paired with a classic IDS covers most techniques and provides a holistic security view.

Figure 7 illustrates the techniques successfully detected by KIDS, represented in green. This figure demonstrates that KIDS focuses on detection in latter stages of attack chain involving operational changes, such as Manipulation of control (T0831), unauthorized command message (T0855), service stop (T0881) and so on.

KIDS demonstrated coverage of multiple high-impact OT specific techniques that are typically missed by traditional IDS solutions. the green highlights in Figure 7 illustrate that KIDS successfully detected process-layer manipulation, even when no conventional network anomaly was visible. For example, KIDS detected malicious logic injection by recognizing changes to the process outputs

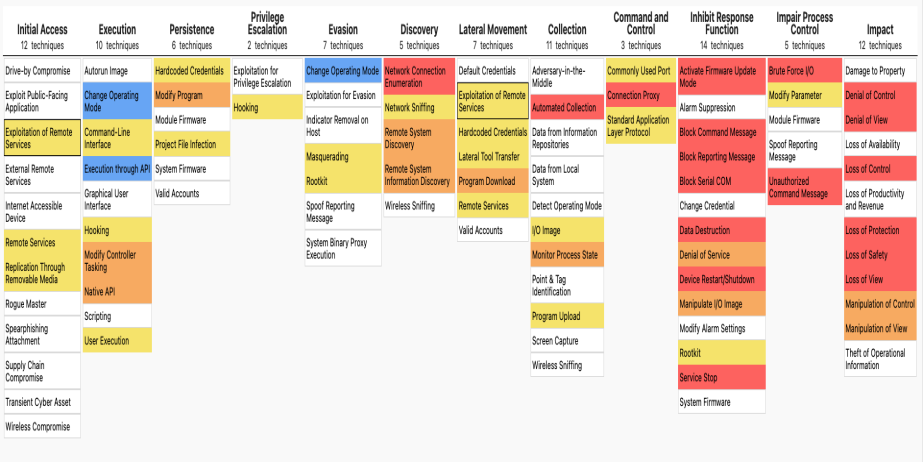


Fig. 5. MITRE ATT&CK Navigator for ICS: ■ Industroyer, ■ Stuxnet, ■ PLC-Blaster, ■ Used by multiple attacks.

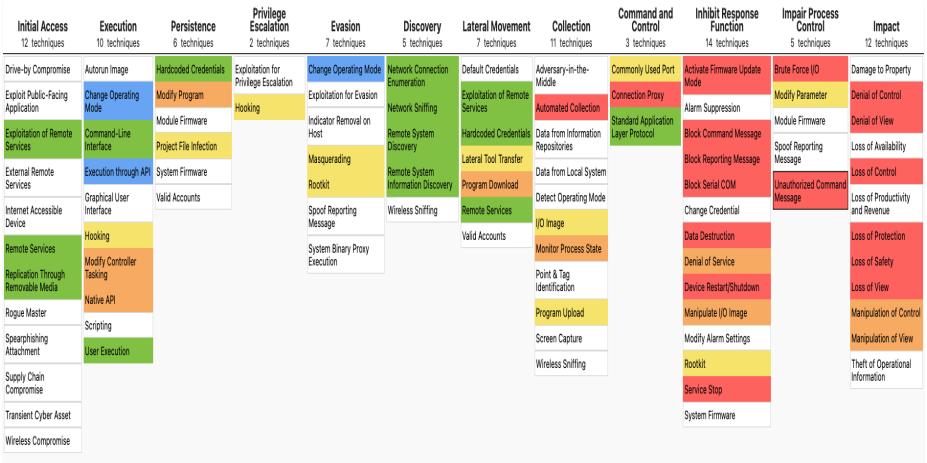


Fig. 6. MITRE ATT&CK Navigator: Techniques from known attacks that might be detected by classic IDS like Snort.

that diverged from annotated process metadata, and flagged Alarm suppression by correlating lack of alarms with breached thresholds. This section demonstrates how KIDS covers techniques that directly affect the physical systems. These are often not detectable by IT focused tools alone, reinforcing the value of process-aware detection, as employed by KIDS.

By using MITRE ATT&CK for ICS as a benchmark, we confirm that KIDS fills a critical gap in the detection landscape, bridging the blind spot between IT centric defenses and OT operational realities.

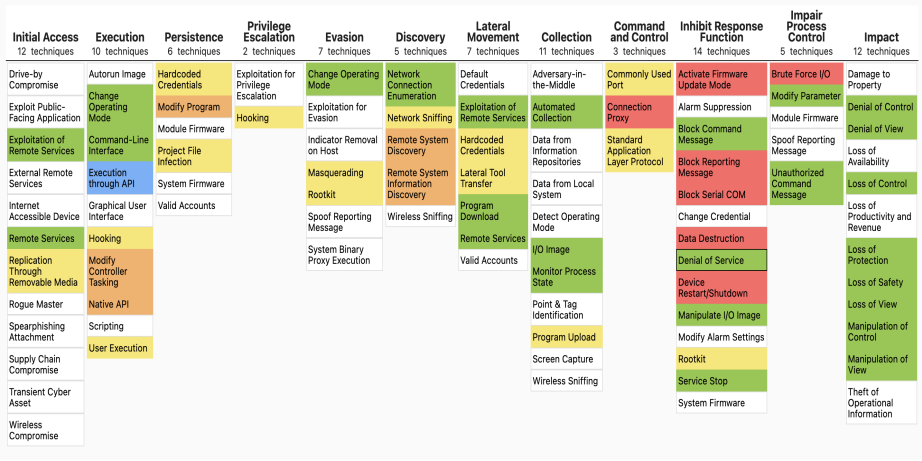


Fig. 7. MITRE ATT&CK Navigator: Techniques from known attacks detected by KIDS.

6 Conclusion and Future Work

This work demonstrates that KIDS significantly enhances ICS threat detection by integrating process aware monitoring with network monitoring. KIDS ensures the adversarial actions at operational level are effectively exposed and detected. A key feature of KIDS is the process metadata file, which contextualizes operational data, enabling security teams to better understand the criticality of operational maneuvers, thereby improving detection accuracy. By annotating critical process variables, such as safety thresholds, system interdependence, KIDS allows for focused precise control over the detection process, reducing false positives improving response times and saving time and resources. This ensures that even if the attacker bypass traditional network defenses, their impact on physical processes remain visible, preventing silent sabotage of critical operations.

One of the fundamental challenges in ICS security is that evasion techniques have significantly evolved in IT environments due to the advancements in technology. However, in OT environments, evasion methods remain relatively stagnant, primarily because control systems are inherently constrained in their execution capabilities. Attackers operating at the control layers do not have access to the same flexibilities provided by sophisticated IT systems. This imbalance creates a unique opportunity for ICS security, while advanced IT solutions struggle to detect advanced evasive threats, process aware monitoring at OT control layers offers more stable and predictable deception landscape. KIDS leverages this advantage by focusing on process integrity, ensuring that even well disguised cyberthreats can be hunted.

The future work aims to further enhance KIDS by automating threat hunting by using automating tools to run the scripts periodically as and when required to make KIDS more flexible. Furthermore, we also aim to integrate Artificial

Intelligence (AI) generated threat hunting hypothesis and hunting scripts using deep learning models. This will enable adaptive and user friendly interface which will eliminate the resource intensive nature of producing scripts manually for hunting threats. This will also add better scope for qualitative analysis of KIDS in future. By bridging IT-OT security gaps, automating threat hunting and detecting in future and leveraging the advances in AI such as Large language Models (LLMs), KIDS aims to set a better standard for industrial cybersecurity ensuring sophisticated adversaries do not remain undetected within such setups including critical infrastructures.

Acknowledgments. This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant number 18/CRT/6222. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Sellafield nuclear site hacked by groups linked to russia and china (Dec 2023), https://www.theguardian.com/business/2023/dec/04/sellafield-nuclear-site-hacked-groups-russia-china
2. Ajmal, A.B., Alam, M., Khaliq, A.A., Khan, S., Qadir, Z., Mahmud, M.A.P.: Last line of defense: Reliability through inducing cyber threat hunting with deception in scada networks. IEEE Access **9**, 126789–126800 (2021). https://doi.org/10.1109/ACCESS.2021.3111420
3. Aldauiji, F., Batarfi, O., Bayousef, M.: Utilizing cyber threat hunting techniques to find ransomware attacks: A survey of the state of the art. IEEE Access **10**, 61695–61706 (2022). https://doi.org/10.1109/ACCESS.2022.3181278
4. Alliance, O.C.: Lateral movement via wmi huntbook. https://github.com/opencybersecurityalliance/kestrel-huntbook/blob/main/huntbooks/Lateral%20Movement%20via%20WMI.ipynb (2022), open Cybersecurity Alliance GitHub Repository
5. Alves, T.R., Buratto, M., De Souza, F.M., Rodrigues, T.V.: Openplc: An open source alternative to automation. In: IEEE Global Humanitarian Technology Conference (GHTC 2014). pp. 585–589. IEEE (2014)
6. Arafune, M., Rajalakshmi, S., Jaldon, L., Jadidi, Z., Pal, S., Foo, E., Venkatachalam, N.: Design and development of automated threat hunting in industrial control systems. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). pp. 618–623 (2022). https://doi.org/10.1109/PerComWorkshops53856.2022.9767375
7. Baezner, M., Robin, P.: Stuxnet. Tech. rep., ETH Zurich (2017)
8. Bhardwaj, A., Goundar, S.: A framework for effective threat hunting. Network Security **2019**(6), 15–19 (2019). https://doi.org/10.1016/S1353-4858(19)30074-1, https://doi.org/10.1016/S1353-4858(19)30074-1
9. Bibi, I., Akhunzada, A., Kumar, N.: Deep ai-powered cyber threat analysis in iiot. IEEE Internet of Things Journal **10**(9), 7749–7760 (2023). https://doi.org/10.1109/JIOT.2022.3229722
10. Farwell, J.P., Rohozinski, R.: Stuxnet and the future of cyber war. Survival **53**(1), 23–40 (2011)

11. Jadidi, Z., Lu, Y.: A threat hunting framework for industrial control systems. IEEE Access **9**, 164118–164130 (2021). https://doi.org/10.1109/ACCESS.2021.3133260
12. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**(1), 1–22 (2019)
13. Lee, R.M., Assante, M.J., Conway, T.: German steel mill cyber attack. Industrial Control Systems **30**(62), 1–15 (2014)
14. Lehman, G.: Cyber-attack against ukrainian power plants
15. Open Cyber Security Alliance: Kestrel threat hunting language documentation. `https://kestrel.readthedocs.io/en/stable/index.html` (2024), accessed: 2025-05-11
16. Research, I.: The thrill of cyber threat hunting with kestrel. `https://research.ibm.com/blog/kestrel-cyber-threat-hunting` (2021), [Accessed: 05-Oct-2023]
17. Shu, X., Araujo, F., Schales, D.L., Stoecklin, M.P., Jang, J., Huang, H., Rao, J.R.: Threat intelligence computing. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. p. 1883–1898. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243829, `https://doi.org/10.1145/3243734.3243829`
18. da Silva, J.P.S., Cunha, E.N.: Supervisory system for hidraulic distribution apparatus scadabr application. In: 2017 IEEE First Summer School on Smart Cities (S3C). pp. 161–164. IEEE (2017)
19. Stouffer, K., Falco, J., Scarfone, K.: Guide to industrial control systems (ics) security. Tech. Rep. NIST Special Publication 800-82 Revision 2, National Institute of Standards and Technology (NIST) (2015). https://doi.org/10.6028/NIST.SP.800-82r2, `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf`
20. Wang, Z.: A systematic literature review on cyber threat hunting (2022)
21. Yazdinejad, A., Kazemi, M., Parizi, R., Dehghantanha, A., Karimipour, H.: An ensemble deep learning model for cyber threat hunting in industrial internet of things. Digital Communications and Networks **9** (09 2022). https://doi.org/10.1016/j.dcan.2022.09.008
22. Yazdinejad, A., Zolfaghari, B., Dehghantanha, A., Karimipour, H., Srivastava, G., Parizi, R.M.: Accurate threat hunting in industrial internet of things edge devices. Digital Communications and Networks (2022). https://doi.org/https://doi.org/10.1016/j.dcan.2022.09.010, `https://www.sciencedirect.com/science/article/pii/S2352864822001857`