Context-Aware Evaluation of Industrial Control Actions: A Modbus/TCP Baseline in the OT Lab

Bruno Salmazo
Liscere
June 2026 · Liscere Technical Report · LTR-2026-01 · v1

Abstract

Industrial control protocols such as Modbus/TCP were designed for reliability in trusted networks and carry no native notion of authorisation or operational appropriateness [1, 2]. An action can therefore be valid at the protocol level, and issued over an authorised channel, yet still be inappropriate for the operational state of the plant. This report introduces the Liscere action-evaluation chain, a protocol-agnostic method that assesses an industrial action along four dimensions: protocol operation, automation artefact, operational context, and policy constraint. We instantiate the chain for Modbus/TCP over a controlled process model in the OT Lab and report an initial feasibility demonstration. Using a control pair that holds the protocol operation, target artefact, and written value constant and varies only a declared maintenance window, we show that the same sensitive configuration write yields an alert in production and an allow during maintenance. Two negative controls confirm that the maintenance window is not a blanket alert suppressor; all four constructed scenarios returned their expected decision and evidence record. We are explicit about what the baseline does not yet measure: detection performance, robustness, and generality across context dimensions and protocols.

Keywords: operational technology security, industrial control systems, Modbus/TCP, context-aware policy, action legitimacy, intrusion detection.

1. Introduction

Operational technology (OT) networks were historically isolated and optimised for availability and determinism rather than for security. Field protocols such as Modbus/TCP reflect that heritage: requests are unauthenticated and unencrypted, and any station able to reach a device may issue a well-formed command that the device will execute [1, 2, 5]. As OT and IT networks converge, this assumption of a trusted perimeter no longer holds.

A recurring lesson from real incidents is that damage need not involve malformed traffic or protocol violations. Stuxnet manipulated controllers using legitimate-looking commands [4, 6], and process-aware attacks issue individually valid actions whose harm only becomes apparent at the level of the physical process [10]. Much of the intrusion-detection literature for ICS focuses on detecting anomalous or malformed traffic [7, 9, 13]. Comparatively little addresses a complementary question: given a well-formed and authorised action, is it appropriate for the current operational context?

This report makes three contributions. First, we define the Liscere action-evaluation chain, which characterises an industrial action along four dimensions and evaluates it against explicit policy to produce a decision and an evidence record (Section 3). Second, we instantiate the chain for Modbus/TCP over a controlled process model in the OT Lab (Section 4). Third, we report a feasibility demonstration built around a control pair and two negative controls (Section 5). We stress that this is a feasibility study, not a performance evaluation; the scope and its limits are stated in Sections 6 and 7.

2. Background and Related Work

Modbus/TCP. Modbus is a request-response protocol in which a client issues function codes such as *write single register* (0x06) or *write multiple registers* (0x10) against a server's address space [1]. The TCP/IP

mapping adds framing but no authentication, integrity, or confidentiality [2]. Authorisation, where it exists, is enforced by network segmentation rather than by the protocol.

Guidance and standards. NIST SP 800-82 [5] and the IEC 62443 series [3] codify defence-in-depth for ICS, including zoning, monitoring, and least privilege. These frameworks motivate, but do not by themselves implement, context-aware evaluation of individual actions.

Detection approaches. Surveys of ICS anomaly and intrusion detection [7, 9, 13] cover signature, specification, and learning-based methods; public Modbus datasets support this work [8], and device fingerprinting characterises endpoints [12]. Physics- and process-aware methods reason about the controlled process to catch actions that are valid on the wire yet harmful in effect [10]. Our work is positioned alongside these as a policy- and context-aware layer: rather than asking whether traffic is anomalous, it asks whether an otherwise valid action is appropriate given a declared operational context.

3. The Action-Evaluation Chain

We model an industrial action and evaluate it along four dimensions. **Protocol operation** is the action at the wire level (for Modbus/TCP, the function code, target address, and written value). **Automation artefact** is the logical asset that the address denotes, for example a high-alarm setpoint exposed as the artefact `ALARM_HI_SP`. **Operational context** is the declared state of the plant, such as whether a maintenance window is in effect. **Policy constraint** is the set of rules that map the preceding dimensions to a decision in {`allow`, `alert`}, together with the evidence recorded for that decision.

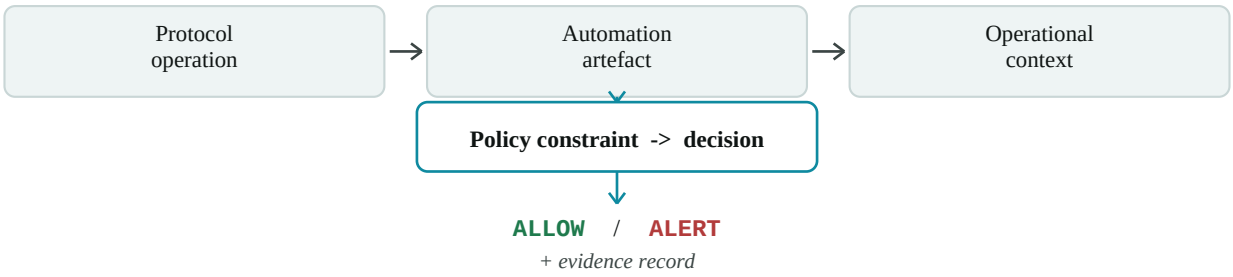

Figure 1: The action-evaluation chain. Three of the four dimensions (protocol operation, automation artefact, and operational context) describe the action; the fourth, policy constraint, evaluates them to yield a decision and an evidence record. The chain is protocol-agnostic, and Modbus/TCP is the first instantiation.

The chain is deliberately protocol-agnostic: the protocol-operation dimension is the only part that is protocol-specific, and it is isolated behind a mapping from raw protocol fields to artefacts. Adding a protocol amounts to providing that mapping, leaving the context and policy machinery unchanged.

4. Experimental Setup

4.1 OT Lab

Experiments run in OT Lab, a controlled laboratory that instantiates the chain and replays constructed scenarios. The laboratory is observational and carries no production enforcement: it emits decisions and evidence but does not block traffic. Source and scenarios are available at `github.com/LiscereSecurity/OT-Lab` [11].

4.2 Process model and artefacts

The process model exposes a small set of registers, including a high-alarm setpoint mapped to the artefact `ALARM_HI_SP`. The model distinguishes in-range from out-of-range setpoint values, and marks selected artefacts as sensitive configuration targets. Address-to-artefact mappings are defined manually for this baseline.

4.3 Policy and evidence

Policy is expressed as explicit rules over the four dimensions and includes a single operational-context dimension, the boolean `maintenance_window`. Each evaluation emits an evidence record identified as `OBS-Rxxx` that captures the action fields, the declared context, the matched rule, and the resulting decision (Appendix A).

5. Evaluation

5.1 The control pair

The control pair isolates operational context as the only independent variable. It holds the protocol operation, target artefact, address, and written value constant, and varies only `maintenance_window`. The same sensitive configuration write to `ALARM_HI_SP` returns two different decisions (Table 1).

Condition	Maintenance window	Decision	Evidence
Production	`false`	**ALERT**	`OBS-R002`
Maintenance	`true`	**ALLOW**	`OBS-R004`

Table 1: The control pair. Identical sensitive write; the declared maintenance window is the only difference.

5.2 Negative controls and scenario matrix

A context dimension that simply suppressed alerts during maintenance would be unsafe. Two negative controls check that the maintenance window is not a blanket suppressor: an out-of-range setpoint write during maintenance still alerts, and a normal in-range write during maintenance still allows. Table 2 lists the full set of four constructed scenarios with their expected and observed decisions.

#	Scenario	Maintenance window	Expected	Decision	Evidence
`S0`	Normal in-range setpoint write	`true`	allow	**ALLOW**	`OBS-R000`
`S1`	Out-of-range setpoint write	`true`	alert	**ALERT**	`OBS-R001`
`S2`	Sensitive write to ALARM_HI_SP	`false`	alert	**ALERT**	`OBS-R002`
`S3`	Sensitive write to ALARM_HI_SP	`true`	allow	**ALLOW**	`OBS-R004`

Table 2: Constructed scenarios. Each row states the declared context, the expected decision, the observed decision, and the evidence record. All four matched.

5.3 Outcome

All four scenarios returned their expected decision and the expected matched rule, with a corresponding evidence record. The chain therefore runs end to end for one protocol, one process model, and one declared operational-context dimension.

6. Discussion

The baseline establishes a modest but specific point: a decision can follow declared operational context rather than protocol validity alone, and constructed scenarios can be checked against explicit expected decisions and evidence records. The maintenance-window case is a deliberately simple instance of a broader idea, that the appropriate response to an identical, valid action may differ with the state of the plant.

The context here is *declared* rather than *inferred*: the laboratory is told whether a maintenance window is in effect. This is a reasonable starting point because many operational states are already recorded in work-order and change-management systems. It also keeps the demonstration interpretable: every decision is traceable to a declared input and a named rule. The chain is intended to complement, not replace, protocol and anomaly

detection within a defence-in-depth posture [3, 5].

7. Limitations

This is a feasibility demonstration and its claims are correspondingly narrow. (i) It covers a single protocol, Modbus/TCP, and a single controlled process model. (ii) Policy rules and address-to-artefact mappings are defined manually. (iii) Operational context is declared, not inferred from traffic or telemetry. (iv) The study does not measure detection performance, false-positive behaviour, robustness to evasion, or generality across additional context dimensions. (v) The laboratory is observational and applies no production enforcement. The results should be read as evidence that the approach is implementable and internally consistent, not as evidence of operational effectiveness.

8. Conclusion and Future Work

We presented the Liscere action-evaluation chain and an initial Modbus/TCP instantiation in the OT Lab. A control pair and two negative controls show that the same valid action can be allowed or alerted depending on a declared operational context, with every decision backed by an evidence record. Future work will broaden the context model beyond a single declared dimension, add further protocols through the protocol-operation mapping, explore inferred rather than declared context, and move from constructed scenarios to a quantitative evaluation with adversarial and false-positive analysis.

References

[1] Modbus Organization. Modbus Application Protocol Specification V1.1b3, 2012.

[2] Modbus Organization. Modbus Messaging on TCP/IP Implementation Guide V1.0b, 2006.

[3] International Electrotechnical Commission. IEC 62443-3-3:2013, Industrial communication networks, Network and system security, Part 3-3: System security requirements and security levels. IEC, 2013.

[4] N. Falliere, L. O. Murchu, and E. Chien. W32.Stuxnet Dossier, version 1.4. Symantec Security Response, 2011.

[5] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson. Guide to Operational Technology (OT) Security. NIST Special Publication 800-82, Revision 3, 2023.

[6] R. Langner. Stuxnet: Dissecting a Cyberwarfare Weapon. IEEE Security & Privacy, 9(3):49-51, 2011.

[7] B. Galloway and G. P. Hancke. Introduction to Industrial Control Networks. IEEE Communications Surveys & Tutorials, 15(2):860-880, 2013.

[8] T. Morris and W. Gao. Industrial Control System Traffic Data Sets for Intrusion Detection Research. In Critical Infrastructure Protection VIII, IFIP AICT 441, pages 65-78. Springer, 2014.

[9] I. Garitano, R. Uribeetxeberria, and U. Zurutuza. A Review of SCADA Anomaly Detection Systems. In Soft Computing Models in Industrial and Environmental Applications (SOCO), Advances in Intelligent and Soft Computing, vol. 87, pages 357-366. Springer, 2011.

[10] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks Against Process Control Systems: Risk Assessment, Detection, and Response. In ACM ASIACCS, pages 355-366, 2011.

[11] Liscere. OT Lab: a controlled laboratory for industrial action evaluation. https://github.com/LiscereSecurity/OT-Lab, 2026.

[12] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah. Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. In NDSS, 2016.

[13] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri. The Cybersecurity Landscape in Industrial Control Systems. Proceedings of the IEEE, 104(5):1039-1057, 2016.

Appendix A. Evidence record

Each evaluation emits an evidence record. The fields below describe record `OBS-R002`, the production half of the control pair: a sensitive write to `ALARM_HI_SP` with no maintenance window, which the policy alerts on.

Field	Value
id	OBS-R002
function_code	0x06 (write single register)
artefact	ALARM_HI_SP (sensitive: true)
value	5
maintenance_window	false
matched_rule	sensitive-write-requires-window
decision	alert