Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 21 July 2026
Revised 25 Aug 2026
Accepted 10 Sep 2026

Natural Resources for Human Health 2026, Vol. 6 (22s): 423–428

KEYWORDS:
Operational technology security; industrial control systems; safety instrumented systems; process safety time; minimum cut; intrusion response.

SafeCut: Deadline-Bounded Safety-Constrained Network Isolation for Preemptive Response in Operational Technology Systems

Austin Minto John[1], Rishabh Jain[1], Dipanjan Das[1], Dr. N. Balaji[1]*

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
*Corresponding author

ABSTRACT: Industrial control systems that govern chemical, energy and water infrastructure increasingly share network infrastructure between information technology and operational technology (OT) domains. When an intrusion is detected, conventional incident response isolates the compromised zone by severing its network links, yet in an OT network the same links often carry the signals of safety instrumented functions (SIFs). The containment action can therefore disable cooling, pressure relief or emergency shutdown and precipitate the physical incident it was intended to prevent. We present SafeCut, a runtime intrusion-response method that isolates a compromised device while formally guaranteeing that every SIF continues to meet its physical response deadline. We model the network as a latency-annotated graph and formulate containment as a deadline-bounded constrained minimum cut, in which the Process Safety Time budget of each loop, rather than its current path, is the binding constraint. An independent verifier removes the proposed cut from a separate graph replica and certifies deadline compliance without trusting the solver. On SafeCut-Bench, a purpose-built corpus of 6,000 scenarios spanning 12 to 400 nodes, blind quarantine preserved timely safety action in only 37.1% of scenarios and static segmentation in 10.2%, whereas SafeCut achieved full deadline compliance in every scenario in which it returned a cut, with no containment failure and a median of one link severed. Deadline budgeting returned a certified response in 47.5% of scenarios that a path-preserving formulation declared infeasible, with no regression in the opposite direction (McNemar $p < 0.001$). A verification suite of 34 independent checks passed in full.

1. INTRODUCTION

Operational technology (OT) refers to the hardware and software that monitors and controls physical processes in industrial facilities, including programmable logic controllers (PLCs), sensors, actuators, supervisory systems and, critically, safety instrumented systems (SIS). For much of their history these networks were physically separated from corporate information technology (IT) infrastructure. The demands of remote monitoring, predictive maintenance and enterprise data integration have progressively dissolved that separation, so that modern industrial networks are converged environments in which enterprise and control traffic share physical infrastructure (Makrakis et al., 2021). This convergence has expanded the attack surface of facilities whose failure modes are physical rather than informational: a compromised database results in data loss, whereas a compromised chemical reactor can result in fire, toxic release or explosion.

The safety instrumented system is the final protective layer in such facilities. It monitors process variables and, when limits are exceeded, commands actions that return the plant to a safe state. Its communication paths carry the sensor readings and actuator

commands on which that protection depends, and those paths have a timing requirement as well as a connectivity requirement. The protective action must reach the final control element within the Process Safety Time (PST), the interval within which intervention still holds the process inside its safe operating envelope (IEC 61511, 2016). Conventional cybersecurity practice responds to a detected intrusion by isolating the compromised host or network zone, a practice transplanted directly from enterprise security, where the cost of isolation is service disruption. In an OT network the same links that carry an adversary's lateral movement may also carry safety instrumented signals, and an isolation action chosen without regard to that fact can disable the plant's protective layer at precisely the moment it is most needed. The TRITON incident of 2017, in which malware deliberately targeted Triconex safety controllers at a petrochemical facility, established that safety systems are now an explicit adversary objective (Johnson et al., 2017; Dragos, 2017).

In this work we present SafeCut, a runtime intrusion-response system that isolates a compromised device in an OT network while formally guaranteeing that every safety-critical control loop continues to meet its physical response deadline. We model the network as a graph in which devices are vertices and communication links are edges carrying measured latency. Rather than preserving a particular communication path, which we show to be over-constrained, we treat the Process Safety Time budget of each safety instrumented function as the binding constraint. A link carrying protective traffic may therefore be severed provided a surviving route still delivers the protective action within its deadline. An independent verification stage then removes the proposed cut from a separate graph replica and confirms by reachability and latency analysis that every protective loop remains within budget, emitting a certificate that is checkable without trusting the solver that produced it. The remainder of this paper describes the system model and methods (Section 2), reports the experimental evaluation on a purpose-built benchmark of 6,000 scenarios (Section 3), and concludes with limitations and future work (Section 4).

2. MATERIALS AND METHODS

2.1 System and Threat Model

We represent the OT network as an undirected graph in which each vertex is a device and each edge carries a measured propagation and switching latency. Edges participating in a safety instrumented function (SIF) are tagged, and each SIF records its sensing element, logic solver, final control element and Process Safety Time budget. The reference configuration comprises twelve devices, of which five are protective, with four registered functions: a cooling loop, a pressure-control loop, an emergency-shutdown loop and an alarm loop. The adversary is assumed to have compromised one or more devices and to be moving laterally through the network. Intrusion detection is treated as a solved input: a policy-based monitor observes Modbus/TCP traffic and raises a containment trigger on an unauthorised write to a protected process register, identifying the source device. Detection quality is not a subject of this investigation.

2.2 Safety-Constrained Minimum-Cut Formulation

On receipt of a containment trigger the solver constructs a flow network in which every safety-critical node is connected to a virtual super-sink by an edge of infinite capacity, ordinary edges are assigned unit capacity, and edges incident on the compromised device are assigned a marginally reduced capacity so that, among cuts of equal cost, the adversary is enclosed rather than the plant being segregated. A minimum cut separating the compromised device from the super-sink is computed using the Edmonds-Karp augmenting-path maximum-flow algorithm (Ford and Fulkerson, 1956; Edmonds and Karp, 1972). Because an augmenting-path procedure cannot saturate an edge of infinite capacity, no safety edge can appear in the resulting minimum cut, so the connectivity of protective loops is preserved by construction.

The central design decision concerns which edges are protected. A formulation that protects every edge presently carrying protective traffic guarantees connectivity but is over-constrained: it declares the problem infeasible in cases where a safe containment exists via a slower but still timely route. We therefore invert the constraint. The solver begins with no protection, computes a cut, and evaluates the post-cut latency of every protective loop. Where a loop still meets its Process Safety Time budget, no protection is added even though its former path was severed; where a budget is exceeded or no route survives, the edges of that loop's fastest route are protected and the computation repeats. The procedure converged within a mean of 1.08 refinement rounds over the evaluated corpus, with a maximum of five. If the compromised device lies inside a protective loop, the method reports the case as infeasible rather than returning an isolation that violates the constraint.

2.3 Independent Certificate Verification

The certificate verifier is decoupled from the solver so that the safety guarantee does not rest on the correctness of the optimisation code. It receives the proposed cut, removes those edges from an independent replica of the topology graph, and for each protective function recomputes, by breadth-first reachability, the end-to-end latency from sensing element through logic solver to final control element. The resulting certificate records, for every function, its post-cut latency, its budget, the margin between them and a deadline verdict. A response is accepted only if every function is within budget. Because verification uses a different method and an independent data structure, a defect in the solver cannot conceal itself.

2.4 Benchmark Construction and Implementation

No public dataset exists for safety-constrained intrusion containment in OT networks; existing ICS security datasets supply process telemetry intended for detection research, not network topology with protective-loop timing constraints intended for response research. We therefore constructed SafeCut-Bench, a corpus of 6,000 independently generated scenarios spanning five topology families (Purdue-layered, redundant ring, star cell, meshed cell and hierarchical tree) and five size strata from 12 to 400 nodes, containing in aggregate 23,183 safety instrumented functions and 1,179,644 network links. Per-link latency is composed of store-and-forward switch delay and link serialization delay sampled by media class, and Process Safety Time budgets are allocated by protective-function class in accordance with IEC 61511 practice, spanning 80 ms for fast pressure protection to 3 s for alarm functions. The corpus is reproducible from recorded seeds and carries a content hash. The system was implemented as a Python core engine using NetworkX and pymodbus with nftables enforcement, together with a web platform presenting a dual attacker and defender console. Validation uses a physics-based continuous stirred-tank reactor (CSTR) model with an exothermic reaction, communicating over Modbus/TCP, so that the physical consequence of a network decision is observable as process behaviour: a preserved cooling loop holds the reactor near its 355 K setpoint, whereas a severed loop proceeds to thermal runaway.

2.5 Statistical Analysis

Four response strategies were executed against every scenario with an identical intrusion and process disturbance, so that the response strategy is the only independent variable: blind quarantine (M1, severing the whole zone of the compromised device), static segmentation (M2, severing the conduits crossing the zone boundary), a rigid path-preserving constrained cut (M3), and the proposed deadline-bounded method (M4). Paired feasibility outcomes between M3 and M4 were compared with the McNemar test; cut sizes were compared with the Wilcoxon signed-rank test; and proportions are reported with 95% confidence intervals. Solver optimality was checked against an exhaustive oracle enumerating all cuts up to three links on a subset of small instances. A verification suite of 34 independent checks, organised into eight families, rebuilt every graph from stored records and recomputed every safety verdict from scratch.

3. RESULTS AND DISCUSSION

3.1 Safety Preservation Across Response Strategies

Table 1 reports the four response strategies across the 6,000-scenario benchmark. The safety result is categorical. Blind quarantine, the prevailing practice, left protective functions able to act within their deadlines in only 37.1% of scenarios (95% CI 35.9 to 38.4%), and static segmentation in only 10.2%. Neither constrained method admitted a single deadline violation across every scenario in which it returned a cut, and neither admitted a single failure of containment. Containment itself is not the distinguishing factor: blind quarantine contains the adversary in every scenario, as it must, since it severs an entire zone, and static segmentation contains in 94.1%. The constrained methods contain in every scenario while severing a median of one link against seventeen and thirty-two respectively. The quantity at stake is therefore not a marginal improvement in a security metric but the difference between a plant retaining and losing its protective capability during incident response.

Table 1. Response-strategy comparison on the 6,000-scenario SafeCut-Bench corpus.

Metric	Blind Quarantine	Static Segmentation	Path-Preserving Cut	SafeCut
Safety deadlines met (%)	37.1	10.2	100.0	100.0
Safety loops connected (%)	37.1	10.2	100.0	100.0

Adversary contained (%)	100.0	94.1	100.0	100.0
Scenarios with a cut returned (%)	100.0	100.0	92.2	95.9
Median links severed	17	32	1	1
Mean links severed	55.01	64.97	1.80	1.82

3.2 Deadline Budgeting Versus Connectivity Preservation

The comparison that establishes the contribution is between the rigid path-preserving cut (M3) and the deadline-bounded method (M4), which differ only in whether the constraint is the path or the deadline. M3 returned no cut in 467 scenarios; M4 returned a certified, deadline-compliant cut in 222 of those same scenarios, a resolution rate of 47.5%, and there was no scenario in which M3 succeeded and M4 failed. A McNemar test on the paired feasibility outcomes gives b = 222, c = 0, a chi-squared statistic of 220.0 and a p-value below 0.001, so the improvement is a strict dominance rather than a trade. As shown in Table 2, the gain is largest on the smallest networks, which offer few alternative routes, and narrows as alternative routes become plentiful, from 6.5 percentage points at the smallest stratum to 1.5 at the largest. On cut size, restricted to the 5,533 scenarios in which both methods returned a cut, M4 severed fewer links in 76 scenarios and more in one, with a Wilcoxon signed-rank p-value of 0.047; the method exploited a permitted reroute, severing a link carrying protective traffic while still meeting the deadline, in 345 scenarios, and in every one the certificate confirmed full deadline compliance.

Table 2. Feasibility by network-size stratum: path-preserving versus SafeCut.

Size stratum (nodes)	Path-Preserving feasible (%)	SafeCut feasible (%)	Gain (pp)
XS (12–24)	81.9	88.4	+6.5
S (25–49)	90.2	94.7	+4.5
M (50–99)	93.9	97.5	+3.6
L (100–199)	96.5	99.1	+2.6
XL (200–400)	98.3	99.8	+1.5

3.3 Solver Scalability and Verification

Solver latency scales with network size, as the polynomial complexity of the augmenting-path method predicts (Table 3). The mean solve time is 0.76 ms at 18 nodes and 14.26 ms at 301 nodes, with a 95th percentile of 68.2 ms and an observed maximum of 150.8 ms on the largest topologies, so the method operates comfortably within a 100 ms response allocation at the scales evaluated. Independent verification is what allows these figures to be treated as evidence rather than assertion: the suite of 34 checks rebuilds every graph from stored records and recomputes every verdict from scratch, and all 34 passed. An exhaustive oracle enumerating every cut up to three links on 427 small instances confirms that the solver never returns a cut smaller than the true optimum, which would indicate an unsound result, and attains that optimum in 99.8% of instances.

Table 3. Solver latency by network-size stratum.

Size stratum	Mean nodes	Mean solve time (ms)	95th percentile (ms)
XS	18.0	0.76	1.46
S	37.2	1.80	2.80
M	74.6	2.57	4.05
L	149.0	5.98	9.50
XL	301.4	14.26	68.21

3.4 Discussion

Across the reviewed literature a consistent gap emerges. Preventive segmentation standards such as NIST SP 800-82 (NIST, 2023) and the ISA/IEC 62443 series (ISA/IEC, 2021) define static zones but offer no runtime decision procedure. Automated response systems act rapidly but indiscriminately, selecting from predefined actions or partitioning along predefined boundaries, with cost models that measure IT disruption rather than the survival of physical protective functions (Rose et al., 2022; Shameli-Sendi and Dagenais, 2015). Optimisation frameworks demonstrate the tractability of formal reasoning over ICS graphs but optimise risk against budget rather than treating protective loops as inviolable (Buczkowski et al., 2022). SafeCut addresses this gap by binding the network decision to a physical time constant of the controlled process. The insight that distinguishes it from the obvious formulation is that preserving the connectivity of a protective path is both insufficient, because a surviving but slower route may deliver the action too late, and over-constrained, because forbidding the severance of any link carrying protective traffic rejects containments that would in fact have been safe.

Three limitations are recorded. First, the lazy protection procedure is sound but not invariably minimal: in one of 427 oracle-verified instances it returned a four-link cut where a two-link cut sufficed. Second, in one of 5,533 paired scenarios the deadline-bounded method returned a cut one link larger than the rigid method, an artefact of tie-breaking between equal-cost partitions. Third, the topologies are synthetic; they follow published architectural patterns and their latency and deadline distributions sit within documented engineering ranges, but they are not traced from a physical facility, and queuing, jitter and congestion are not simulated. None of these limitations affects the safety guarantee, which held categorically across every scenario: no cut admitted by the certificate gate ever caused a protective deadline to be missed. Establishing transfer to a physical installation requires evaluation on an industrial testbed, which places the current work at technology readiness level four.

4. CONCLUSION

We addressed a problem acknowledged in the OT security literature but for which no rigorous solution had been proposed: conventional incident response, applied to an industrial network, can disable the plant's protective layer while containing the adversary, and thereby cause the physical incident it exists to prevent. Our contribution is to formulate containment as a deadline-bounded constrained minimum cut, in which the Process Safety Time budget of each safety instrumented function, rather than its current path, is the binding constraint, paired with an independent certificate that confirms deadline compliance without trusting the solver. On a purpose-built corpus of 6,000 scenarios, blind quarantine left protective functions able to act within deadline in 37.1% of scenarios and static segmentation in 10.2%, whereas our method achieved full deadline compliance in every scenario in which it returned a cut, with no containment failure and a median of one link severed. Deadline budgeting returned a certified response in 47.5% of the scenarios that the path-preserving formulation declared infeasible, with no regression in the opposite direction. Future work follows directly from the stated limitations: evaluation on a physical testbed with real PLCs to reach technology readiness level five, extension to the case of in-loop device compromise, and incorporation of queuing and congestion into the latency model so that the deadline constraint holds under load rather than only under nominal traffic.

Acknowledgements

The authors acknowledge the School of Computer Science and Engineering, Vellore Institute of Technology, for the infrastructure and guidance provided during this project, and thank the faculty of the school for their feedback during the evaluation phase.

Author Contributions

Author 1: Conceptualization, Methodology, Software, Writing, Original Draft. Author 2: Software, Validation, Data Curation, Visualization. Guide Name: Supervision, Conceptualization, Writing, Review and Editing.

Conflict of Interest

The authors declare that there is no conflict of interest.

Ethics Statement

Ethical approval was not required for this study. The work involves no human participants, animal subjects or personally identifiable data. All experiments were conducted on synthetic network topologies and a simulated process model.

Data Availability Statement

The data supporting the findings of this study, including the SafeCut-Bench corpus generator, the recorded random seeds, the solver and verifier implementations and the verification suite, are available from the corresponding author on reasonable request.

REFERENCES

[1] Buczkowski, P., Malacaria, P., Hankin, C., Fielder, A., 2022. Optimal security hardening over a probabilistic attack graph: a case study of an industrial control system using the CySecTool tool. In: Proceedings of the ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS 22), Baltimore, MD, pp. 1–10.

[2] Dragos, 2017. TRISIS malware: analysis of safety system targeted malware. Dragos Technical Report, Dragos Inc.

[3] Edmonds, J., Karp, R.M., 1972. Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19(2), 248–264.

[4] Ford, L.R., Fulkerson, D.R., 1956. Maximal flow through a network. Canadian Journal of Mathematics 8, 399–404.

[5] IEC 61511, 2016. Functional safety: safety instrumented systems for the process industry sector. International Electrotechnical Commission, Geneva.

[6] ISA/IEC, 2021. ISA/IEC 62443: industrial automation and control systems security. ISA Standards Series, International Society of Automation.

[7] Johnson, B., Caban, D., Krotofil, M., Scali, D., Brubaker, N., Glyer, C., 2017. Attackers deploy new ICS attack framework TRITON and cause operational disruption to critical infrastructure. FireEye/Mandiant Threat Research.

[8] Makrakis, G.M., Kolias, C., Kambourakis, G., Rieger, C., Benjamin, J., 2021. Vulnerabilities and attacks against industrial control systems and critical infrastructures. arXiv preprint arXiv:2109.03945.

[9] NIST, 2023. Guide to operational technology (OT) security. NIST Special Publication 800-82 Rev. 3, National Institute of Standards and Technology, Gaithersburg, MD.

[10] Rose, J.R., Swann, M., Grammatikakis, K.P., Koufos, I., Bendiab, G., Shiaeles, S., Kolokotronis, N., 2022. IDERES: intrusion detection and response system using machine learning and attack graphs. Journal of Systems Architecture 131, 102722.

[11] Shameli-Sendi, A., Dagenais, M., 2015. ORCEF: online response cost evaluation framework for intrusion response system. Journal of Network and Computer Applications 55, 89–107.