AI Gives Attackers OT Expertise on Demand

What the Technique Looks Like on a Real PLC, and What It Costs to Run

Version 1 — April 2026
Don C. Weber
Cutaway Security, LLC
https://www.cutawaysecurity.com
https://www.linkedin.com/in/cutaway/

Table of Contents

1. Executive Summary

I fed a production AI tool the kind of material a threat actor walks away with after compromising an engineer workstation. Ladder logic screenshots. A Modbus address export. A few manufacturer reference PDFs that are already sitting on the vendor's public documentation site. That is all it took.

The AI built a working understanding of the simulated process, mapped the cross-subroutine dependencies that determine how the PLC behaves under scan, and handed back ranked attack paths with protocol-level instructions for executing each one. I tested every path it produced against the physical kit. Two reproduced cleanly. Those two are the same attack chains we teach in SANS ICS613.

The whole analysis cost me about one analyst workday and less than two dollars in application programming interface fees. That is the economic picture for an adversary preparing a targeted attack against an operational industrial process using only stolen or leaked configuration files. The cost of entry is not the constraint, and it was never going to be. The constraint used to be domain expertise, and AI tools now supply that on demand.

Leadership takeaways

Configuration files are process intelligence. Protect them the way you protect safety documentation. Validate the backups. Run integrity checks. Inventory every copy of every file, wherever it lives. That includes copies held by vendors and contractors, and copies that made their way over to the information technology network.

Remote access to engineering workstations deserves your highest-tier scrutiny. The engineer workstation is where the intelligence lives. The ladder logic. The project files. The passwords for the controllers. That endpoint is a high-value target for an adversary who wants to do exactly what this paper demonstrates.

The investment required to weaponize this is trivial. Assume the technique is already being run against configuration files that have left your environment. Size your defender investment against that reality, not a hypothetical one. An underfunded operational technology cybersecurity program is not saving money. It is subsidizing the attack.

The human validation gap favors the attacker. Several attack paths the AI produced did not reproduce cleanly in the lab. In a defender context, that is a failure rate. In an adversary context, it is two successful attacks and some noise. An attacker runs everything and keeps what works. A defender cannot afford that batting average.

2. Scenario and Setup

2.1 The threat model

The scenario here is the one CISA described in advisory AA26-097A, on Iranian-affiliated activity against programmable logic controllers in United States critical infrastructure. It is also the scenario behind the FrostyGoop incident that preceded it. A threat actor gets to an engineering workstation. They pull the configuration files for whichever controllers are reachable from there.

Most organizations do not treat those files as process intelligence. They treat them as operational paperwork, which means limited protection on the workstation and limited protection in the backups. Vendors, integrators, and contractors hold copies too, often long past the end of the project, in locations the operating organization stopped tracking years ago.

Up until recently, those files were not as useful to an attacker as they look on paper. Reading ladder logic takes training. Understanding the physics of an industrial process takes more training. Writing exploit code that speaks the right industrial protocol takes still more training. A threat actor with strong general cybersecurity skills and no industrial background could pull the files all day and still not know what to do with them. That is why precision attacks against operational processes used to be the domain of nation-state actors and a small number of well-resourced criminal groups.

That barrier is gone. A production AI tool supplies the domain knowledge, the protocol knowledge, and the exploit construction on demand, working from the artifact trail on the engineer workstation plus a few manufacturer reference documents that are sitting on the vendor's public documentation site right now.

2.2 The inputs

The full set of materials that produced the analysis described in this paper:

- Twelve PNG screenshots of ladder logic from the CLICK programming software, covering the main program and every subroutine
- One CSV file with 224 Modbus address rows representing process setpoints and configurations
- Ten PDF reference documents from Automation Direct's public documentation site, totaling 48 pages

Nothing proprietary beyond what sits on the engineer workstation. Nothing that required network access to the programmable logic controller. Nothing that a threat actor could not assemble from a compromised engineer workstation plus a few minutes of public web research.

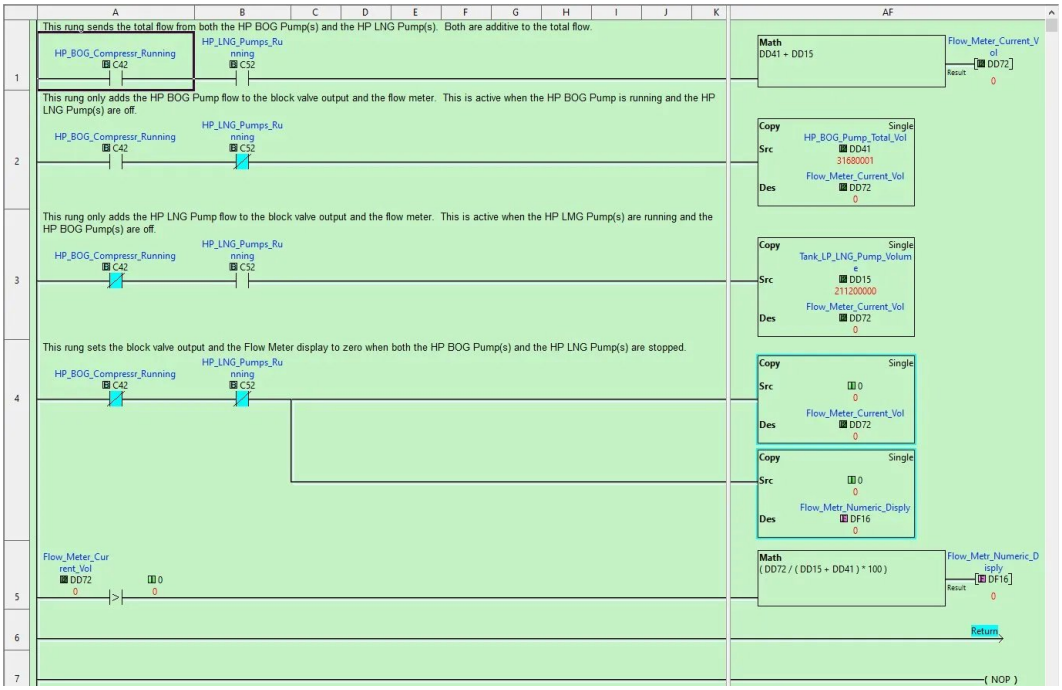

Figure 1. A representative ladder logic subroutine from the artifact trail. This is the Flow_Meter_Control subroutine, shown as exported from the CLICK programming software. Images like this one were the primary input to the AI tool.

A note about process logic files

The CLICK programmable logic controller does not export ladder logic in human-readable configuration files. The CLICK configuration file is a binary blob, which forces analysts to generate images of the ladder logic diagrams. Other controllers and devices allow the exporting of their logic in Extensible Markup Language files. It is significantly easier to analyze logic in an XML file than in a PNG image of ladder logic. The artifact trail available to an adversary targeting a different vendor may therefore be even richer than the trail used in this demonstration.

3. What the Technique Produces

I am going to stay at the capability level in this section. The specific findings are material that SANS ICS613 students work through during the course, and preserving the value of those hands-on exercises matters more to me than publishing the details here. The capability descriptions below are enough to understand what an adversary running this technique will produce against a process of their choosing.

3.1 Reading physics out of the math

The AI knew the simulator was modeling an LNG terminal. I told it that up front. What it did not have was any interpretation of the ladder logic, any explanation of what each subroutine did, or any guidance on how the registers and coils related to each other. All of that had to come from the PNG images of the logic and the CSV of Modbus addresses.

One of the subroutines contains an arithmetic expression that derives tank temperature from tank pressure. The AI read the math, connected the coefficients to the auto-refrigeration behavior of cryogenic storage, and explained the physical relationship correctly. It pulled the LNG context I gave it together with the mathematical shape of the formula, and produced the right physics interpretation. That is process knowledge lifted straight out of PLC code with the correct domain applied to it.

The broader pattern is what matters for defenders. An AI tool with the process domain as context will correctly interpret the physics encoded in the controller's arithmetic. An adversary working against a different vendor, a different process, and a different industrial sector will get similar recognition on whatever physics happens to live in that controller.

3.2 Tracing signals across subroutines

The most operationally useful single output produced during the analysis was not any individual subroutine interpretation. It was the cross-subroutine map: which bit coils are written in one subroutine and consumed in another, which registers cascade effects across multiple logic paths, where scan order determines whether the program's behavior is safe or unsafe.

Building this map by hand is exactly the kind of analysis that takes an experienced controls engineer meaningful time. The AI tool built it from twelve images in a small fraction of that time. More importantly, the map identified single-write-point attacks that affect multiple subsystems simultaneously, which is the kind of attack that operators are least likely to detect because the observable effects are distributed across several parts of the human-machine interface.

3.3 Dead code with live Modbus addresses

One of the subroutines in the project is present in the program file but is never actually called from the main program during normal operation. The registers that this subroutine would have written are still present in the Modbus address export, with descriptive nicknames that imply they hold live process values. A monitoring tool or a human-machine interface built from the address export would read those registers and present stale values to operators as current process telemetry.

The AI tool identified this condition by walking the main program's call chain and noticing that the subroutine was not in it. That is a relatively subtle analysis step. A human analyst unfamiliar with the

specific project structure would likely miss it on a first pass. The implication for defenders is that address exports on their own do not reliably indicate which registers are genuinely in use at runtime, and monitoring strategies that assume they do are exposed.

3.4 Attack paths with protocol-level instructions

The AI tool produced attack paths not in abstract terms, but as specific write operations against specific addresses with specific values, ordered by operational impact. The output included the expected physical effect on the simulator, the persistence of the change under normal PLC scan behavior, and the recovery conditions required to return the system to its defined starting state. The protocol-level instructions were correct Modbus function codes and address encodings, produced without any protocol expertise on the analyst's part.

4. Validation Against the Physical Kit

4.1 The two attacks that worked

Every attack path produced by the AI tool was tested against the physical ICS613 student kit. Two of the attack chains reproduced cleanly on the hardware. The process responded exactly as the AI tool predicted. Physical light-emitting diodes on the board changed state. The tank display moved. The flow meter reported what the AI said it would report. The kit was no longer reflecting reality, and an operator watching the board had no way to know that the observable state was divergent from the actual process state.

Those two attack chains are the same two that are used as hands-on exercises in the SANS ICS613 penetration testing course, where students walk through them under instructor guidance as part of the curriculum. The AI tool, working from twelve screenshots and a CSV, independently landed on the same attack paths the course teaches. That cuts two ways. It validates that the course exercises reflect what a capable analyst would identify from the artifact trail, and it validates that the AI's analysis was technically accurate where accuracy mattered most.

4.2 The paths that did not reproduce

Other attack paths identified by the AI tool were less successful when tested against the physical kit. Some had correct logic but depended on timing or state conditions that the AI could not observe from static analysis of ladder logic images. Some predicted behavior that testing showed did not quite match. That outcome is the present reality of AI-assisted analysis of industrial control systems. The AI tool will provide examples of opportunities that may be missed during manual analysis. Sometimes those opportunities work, and sometimes technical and physical limitations prevent the predicted outcome.

The asymmetry that this creates between attacker and defender is discussed in Section 5.3. The short version: the AI's imperfect batting average is an operational reality for defenders, and a non-problem for attackers.

5. The Economics

Defenders and leadership frequently ask some version of the same question after seeing a demonstration like this one. The phrasing varies. The substance is always the same: how realistic is it for an attacker to actually do this? The answer is contained in three numbers. How long it takes, how much it costs, and how much of it an untrained analyst can accomplish without a physical target to validate against.

5.1 Where the time went

The analysis took approximately fifteen to eighteen hours of analyst effort spread across multiple working sessions. That is a full analyst workday and then some.

The analyst profile matters for interpreting that number. The analyst had deep familiarity with industrial control system equipment, with the Modbus and EtherNet/IP protocols, and with the operations of the simulated LNG terminal process. The analyst had limited prior experience reading ladder logic and was a breaker rather than a builder. A significant portion of the time went to learning the visual grammar of ladder logic on the job, and to understanding how the AI tool was describing the process. A controls engineer who already reads ladder logic fluently would compress the same work to eight to twelve hours.

Out of that fifteen to eighteen hour total, the time broke down roughly as follows:

- Five to six hours: initial image capture, interpretation, and understanding each rung's logic
- About four hours: writing up the AI analysis in a structured, reusable format that works as a repeatable AI analysis framework
- About three hours: cross-referencing every mathematical interpretation and Modbus address against the CSV export to verify accuracy
- The remaining three to four hours: understanding what the AI tool produced, and working through testing assumptions against the ICS613 student kit

None of those activities are unique to this analyst or to this kit. An adversary working with a similar artifact trail from a different PLC would spend their time in roughly the same places.

5.2 Where the tokens went

The full analysis consumed approximately 212,000 tokens in total, split between input and output. At the application programming interface rates published at the time of this writing, that worked out to roughly the following costs across the leading production models:

Model	Input rate	Output rate	Estimated cost
Claude Sonnet 4.6	$3.00 / MTok	$15.00 / MTok	~$1.40
GPT-4o	$2.50 / MTok	$10.00 / MTok	~$1.01
GPT-4o mini	$0.15 / MTok	$0.60 / MTok	~$0.06
Claude Opus 4.6	$15.00 / MTok	$75.00 / MTok	~$6.97

GPT-4o mini is cheaper by a factor of approximately twenty, with meaningfully weaker output quality but with enough capability to handle portions of the job under close supervision. For single-subroutine analysis with an experienced analyst reviewing the output, the smaller model is viable. For cross-subroutine synthesis and ranked attack path production, the larger models are needed.

Smaller scopes reduce cost substantially. A single subroutine analysis lands at approximately twenty thousand tokens, for well under fifteen cents. A three-subroutine cluster analysis with a cross-reference summary lands at approximately forty-five thousand tokens, for about thirty cents. The implication is that the economic barrier to iterative, long-running analysis of a target environment is effectively absent.

This is the number that changes the threat conversation. The cost of entry is not the constraint, and it was not going to be. Even if the model were ten times more expensive, the analysis would still run under twenty dollars. The economic barrier to this technique does not exist.

5.3 Where the human still mattered

The human validation gap described in Section 4.2 deserves explicit discussion because it is where defender intuition most often leads to the wrong conclusion. The AI tool produced attack paths based on static analysis of the logic it was given. Static analysis does not capture timing behavior. It does not predict how comparison-driven coils respond to specific write sequences. It does not anticipate how physical analog inputs override PLC register writes during normal scan cycles. Some of what the AI predicted was technically correct logic that the hardware simply did not behave the way the logic suggested.

The analyst caught those mismatches by running each proposed attack against the physical board. An adversary without a physical target to validate against will ship more attack paths than they should, including ones that fail.

The part that defenders need to sit with is this: the adversary does not need a high batting average. One attack path that reproduces cleanly in production is enough to cause operational impact. The AI tool gave the analyst two attacks that worked and several that did not. In a defender context, that is a failure rate. In an adversary context, it is two successful attacks and some noise. The asymmetry favors the attacker.

The human is still in the loop, still mattering, still catching things the AI tool misses. That does not mean the AI threat is less real. It means AI plus a human produces even better results, and there will always be humans willing to iterate against live targets to validate and ensure intended outcomes.

6. The Knowledge Gap Collapsed

Give the AI the process context and the artifact trail, and it does the rest. I told it the simulator modeled an LNG terminal. I did not tell it how the ladder logic worked, what each subroutine did, which registers mattered, or how the scan order affected the behavior. All of that came out of the images and the CSV. The AI pulled the interpretation together from what it already knows about LNG process behavior, about ladder logic grammar, about Modbus addressing, and about the design patterns that show up in controller code across every industrial vendor. It did not need me to teach it any of that.

That is the shift. The old barrier was that a generalist cybersecurity analyst could get their hands on configuration files and still be stuck, because they did not have the domain to read them or the protocol knowledge to act on them. The AI supplies both. What the analyst brings is the artifact trail and enough context to orient the analysis. That is a small set of requirements.

What it needed was the artifact trail. Ladder logic screenshots. A Modbus address export. A few manufacturer reference documents already sitting on the public internet. The kind of materials that live on an engineer workstation, in a vendor's project folder, on a backup share that nobody audits, in an email attachment that somebody forwarded to a contractor three years ago and has not thought about since.

The skill profile required to execute this is not nation-state. It is not even advanced persistent threat. It is a moderately capable security analyst with a laptop. The reason this technique has not yet been observed in the open at scale is not that it is hard. It is that the artifact trail has to get out of the operational technology environment first, and most of the industry still treats that artifact trail as operational paperwork rather than process intelligence.

That window is closing.

For an attacker with cybersecurity skills but no industrial background, the artifact trail is not a closed door. It is a key.

7. Defender Recommendations

The recommendations below combine the actions covered in the blog post series that preceded this paper with the prioritization implications that follow from the economic analysis in Section 5. None of what follows is new. It all matters more now than it did last year.

7.1 Treat configuration files as process intelligence

The ladder logic, the address map, the tag database, the human-machine interface project file. All of it. This material is the operational process expressed in text and numbers, and it must be protected the way safety documentation is protected. That means integrity checks, version control with audit trails, offline backups that an attacker with domain credentials cannot overwrite, and inventories that list where every copy lives.

The inventory requirement deserves emphasis. In most organizations, configuration files exist on the production engineering workstation, on the backup system, on at least one file share, in at least one email chain, and on the laptop of at least one integrator or vendor. Any copy the organization does not have an inventory entry for is a copy the organization cannot protect.

7.2 Keep OT data on the OT side

Configuration files belong on the engineering workstations that need them, not on the shared drive the information technology team lifted into Microsoft 365 to make collaboration easier. Every time a process artifact crosses the boundary toward the enterprise, the operating organization should assume it has left their control. This applies equally to the information that vendors and integrators move in and out of operational environments. Attackers do not have to target the operating organization directly. They know who does the organization's work for them, and they target them.

7.3 Monitor remote access

The engineer workstation with the vendor software installed is where process logic lives, where project files live, where the programmable logic controller passwords live. That endpoint is a high-value target in the operational technology environment for an attacker who wants what AI tools can now analyze. Remote access is one of the SANS Five Critical Controls for industrial cybersecurity for a reason. The demonstration in this paper sharpens that reason: the intelligence that remote access can now produce, once the files are out, is more actionable than it has ever been.

7.4 Validate process logic file backups

AI tools make it straightforward for threat actors to identify which files are most valuable to a process environment. That identification can be as simple as a file extension, for example the .ckp extension used for CLICK programmable logic controllers. AI-developed tools and malware, including ransomware, can now be specifically focused on the key project file assets that are critical for operations. Defenders must anticipate that these attacks are going to be successful, and operators must be practiced at ensuring rapid recovery. Backup validation is not a checkbox. It is a periodic drill.

7.5 Cost arguments no longer hold

The most important implication of the economic analysis in Section 5 is that cost arguments against operational technology cybersecurity investment do not hold up. If the adversary cost of this technique is under two dollars, the defender cost of reasonable mitigation cannot honestly be argued as too expensive relative to the threat. Leadership that declines to fund configuration file protection, remote access monitoring, or penetration testing on cost grounds is making a risk decision, not a cost decision. Those are different conversations and should be framed accordingly in budget meetings.

7.6 Detection and response matter more than prevention

The artifact trail has already left the building in most environments. The question is no longer how to prevent the exfiltration that already happened. The question is how to detect the remote access and the protocol-level manipulation that uses what was exfiltrated. That is what the SANS Five Critical Controls are structured to address, and it is what SANS ICS613 teaches students to assess.

7.7 Assess what you have exposed

The methodology described in this paper is not secret. An adversary reading this paper and the blog series that produced it has most of the approach already. The parts that were held back are the specific attack procedures, and those are not difficult to reconstruct for someone motivated enough. If an operational technology monitoring posture is not presently prepared to see a protocol-level manipulation attempt, that posture needs to move up the priority list. The SANS ICS613 course covers the full penetration testing methodology that this paper sits inside.

8. A Note for Leadership

> **To operational leadership**
> Your engineers know which systems hold the configuration files. Your security team knows whether those systems are monitored. The gap between the two is where the risk is sitting, and closing that gap is not a new project. It is a priority call. Back your teams. Fund the assessment. Get the training. Threat actors are not waiting for the industry to understand AI tools.

The numbers in this paper should settle a question that most leadership teams have been asked in budget meetings at least once already. How realistic is it for attackers to actually do this? At under two dollars per analysis and a workday of analyst time, it is as realistic as any other commodity attacker technique. The question is no longer whether adversaries will use AI tools to understand a target process. The question is whether the defending team will be ready to detect and respond when they do.

The three priority shifts described in the executive summary are not preferences. They are the minimum adjustments required to keep defender investment in proportion to the threat that AI-assisted analysis of stolen configuration files now represents. An underfunded operational technology cybersecurity program in this environment is not saving money. It is subsidizing the attack.

9. References and Further Reading

Cited sources

- CISA Advisory AA26-097A: Iranian-Affiliated Cyber Actors Exploit Programmable Logic Controllers Across US Critical Infrastructure. https://www.cisa.gov/news-events/cybersecurity-advisories/aa26-097a
- MITRE ATT&CK Campaign C0041: FrostyGoop Incident. https://attack.mitre.org/campaigns/C0041/
- SANS Institute: The Five ICS Cybersecurity Critical Controls. https://www.sans.org/white-papers/five-ics-cybersecurity-critical-controls
- SANS ICS613: ICS/OT Penetration Testing and Assessments. https://www.sans.org/cyber-security-courses/ics-ot-penetration-testing-assessments
- SANS ICS Security Summit 2026. https://www.sans.org/cyber-security-training-events/ics-security-summit-2026

Companion blog posts

This paper consolidates two blog posts published on the Cutaway Security blog. Both are available at https://www.cutawaysecurity.com/blog/. The first post covers the demonstration at the capability level. The second post covers the economics of the technique. This paper contains the consolidated version of both posts plus the appendices below.

Background post

The context for this demonstration was established in an earlier post titled "AI Gave Attackers Something We Weren't Ready For. Here's What OT Defenders Need To Do About It." That post frames the broader industry conditions that make this technique worth demonstrating. https://www.cutawaysecurity.com/blog/ai-gave-attackers-something-we-werent-ready-for/

Appendix A: Methodology Notes

Input materials detail

Material	Format	Quantity	Role in analysis
Ladder logic exports	PNG images	12 files	Primary analysis source; covers main program and all subroutines
Modbus address export	CSV	224 rows	Address, nickname, function code, and retentive-flag reference
CLICK data type reference	PDF	1 page	Data type grounding for register interpretation
CLICK memory address reference	PDF	2 pages	Address range reference by register type
CLICK system control relay reference	PDF	8 pages	System bit definitions consumed by the logic
CLICK system data register reference	PDF	16 pages	System register definitions
CLICK Modbus function code reference	PDF	3 pages	Supported Modbus function codes
CLICK network documentation	PDF (several)	~18 pages	Protocol and interface configuration context

Analysis workflow

The analysis was conducted as a series of focused sessions against individual subroutines, with a separate synthesis phase to build the cross-subroutine map. Each subroutine session produced a structured analysis document using a consistent format established early in the work. The consistent format allowed the cross-subroutine synthesis to proceed efficiently because every subroutine document had the same sections in the same order.

Validation against the physical kit occurred after the synthesis phase. The kit was placed in a defined initial state for each test. Each attack path was executed using a standard Modbus client. The physical board's response was observed and recorded. Recovery was performed before the next test.

Known limitations

The analysis has several limitations that defenders should be aware of when interpreting the results:

- Ladder logic image capture artifacts occasionally wrapped variable names in ways that required manual reconstruction. This affected a small number of addresses and was resolved by

cross-reference to the CSV export

• Timer current-value addresses are not present in the CSV export and were derived from CLICK documentation. Accuracy was verified but not exhaustively tested across all timer instances

• Static analysis does not capture PLC scan-cycle timing behavior. Several AI-predicted attack paths failed in the lab for this reason

• The simulator is a training platform and makes deliberate simplifications. Real LNG terminal logic is more complex and would likely produce a larger set of attack paths with a correspondingly larger set of failure modes

Appendix B: Workshop Sizing for Instructors

Instructors considering similar exercises for student cohorts should use the scope tiers below as a planning reference. Each tier reflects an achievable workshop outcome against a defined token budget and session structure. The per-student costs assume direct application programming interface access at Sonnet 4.6 rates.

Workshop tiers

Tier	Scope	Student time	Tokens / student	Cost / student
Tier 1	Single subroutine analysis	45 to 90 min	~20,000	~$0.11
Tier 2	Three-subroutine cluster	3 to 5 hrs	~45,000	~$0.29
Tier 3	Full program analysis	12 to 20 hrs	~190,000	~$1.40

For a 20-student cohort, the aggregate cost sits between two dollars and twenty cents for a Tier 1 workshop and twenty-eight dollars for a Tier 3 capstone. Cost is not the constraint for workshop planning. Instructor availability and session scheduling are the actual constraints.

Factors that reduce per-student time

- Prior programmable logic controller experience: reduces total time by 35 to 45 percent
- Pre-built analysis templates supplied by the instructor: reduces documentation time by 15 to 20 percent
- Access to the controller programming software during the workshop: reduces validation time by 20 to 25 percent
- Starting from an existing system overview document rather than building one: reduces orientation time by 10 to 15 percent

Session planning recommendation

Tier 1 fits cleanly in a single session. Tier 2 can complete in one long session but benefits from being split across two. Tier 3 should be planned as a multi-session effort spread across several days, with instructors briefed that context-window management will be part of the student experience. Students should be advised to save their output documents incrementally rather than waiting until the end of the conversation to export everything at once.

Model selection guidance

Claude Sonnet 4.6 and GPT-4o are roughly equivalent in capability and cost for this task at Tier 2 and Tier 3 scope. GPT-4o mini is significantly cheaper but requires structured prompting support and instructor review of outputs. It is appropriate for Tier 1 introductory exercises with instructor oversight. It is not recommended for independent Tier 2 or Tier 3 work where cross-subroutine synthesis is part of the scope.

About the Author

Don C. Weber is an industrial control systems and operational technology cybersecurity consultant, penetration tester, and instructor. He is the founder of Cutaway Security, LLC, a veteran-owned consultancy specializing in industrial control system security assessments, penetration testing, and SANS training. Don is a co-author of SANS ICS613: ICS/OT Penetration Testing and Assessments, where all author roles are considered equal. He is also a longtime SANS certified instructor for ICS410: ICS/SCADA Security Essentials.

Don holds the ISA/IEC 62443 Cybersecurity Expert certification and has spent more than two decades working in and around industrial environments. His consulting work focuses on penetration testing, gap assessments, and cybersecurity program development for critical infrastructure operators.

Contact and further writing:

- Cutaway Security blog: https://www.cutawaysecurity.com/blog/
- LinkedIn: https://www.linkedin.com/in/cutaway/
- Cutaway Security, LLC: https://www.cutawaysecurity.com
- GitHub: https://github.com/cutaway-security

Go forth and do good things.