ISSN: 2473-4901
v12 n6649

Teaching AI Cybersecurity Risk in AIS: A Control-Mapping Framework and Classroom Application

Jose Victor Lineros
jose.lineros@unt.edu
University of North Texas
Denton, TX 76203

Abstract

Artificial intelligence (AI)-enabled applications are entering accounting transaction cycles, but traditional accounting information systems (AIS) courses provide limited guidance for translating model and large language model (LLM) risks into controls and audit work. This teaching paper presents an AIS control-mapping framework that uses a stable analytical sequence: system boundary and lifecycle stage, vulnerability mechanism, accounting-process consequence, AIS control objective, control activity and owner, and audit evidence, test, and residual risk. Categories from the Open Worldwide Application Security Project (OWASP) Top 10 for LLM Applications 2025 serve as current examples; they are not treated as an exhaustive taxonomy of AI risk. A fictional expense-review case demonstrates the framework within accounts payable, and the classroom design reuses the mapping in short exercises, a case deliverable, an instructor key, and a rubric. The author used an earlier version of the AI-enabled expense-review assignment (not the complete six-field framework and current implementation materials presented here) on two occasions with 158 graduate information technology (IT) audit students in total; because the evidence consists only of informal instructor observations, no causal learning claim is made. By connecting AI application vulnerabilities to authorization, validity, accuracy, segregation of duties, audit trails, evidence reliability, and governance, the framework enables instructors to add a focused AI cybersecurity unit to existing AIS or IT audit courses without requiring model-development expertise.

Keywords: AIS Education, Artificial Intelligence, Cybersecurity Risk, Internal Control, IT Audit

Teaching AI Cybersecurity Risk in AIS: A Control-Mapping Framework and Classroom Application

Jose Victor Lineros

1. INTRODUCTION

Artificial intelligence is increasingly embedded in transaction processing, anomaly detection, financial analysis, and audit support (Sutton et al., 2016; Vasarhelyi et al., 2015). These uses can improve speed and coverage, but they also change what an evaluator must inspect. A conventional application may be assessed through configured rules, program changes, access rights, and reconciliations. An AI-enabled application can also depend on training data, model versions, natural-language instructions, third-party components, and probabilistic outputs. The Open Worldwide Application Security Project (OWASP) is a nonprofit foundation that works to improve software security. Its guidance for LLM applications highlights how a process may satisfy familiar access-control requirements while still authorizing an invalid transaction, disclosing sensitive data, or producing evidence whose provenance cannot be established (OWASP Foundation, 2024).

These issues belong in AIS because they affect accounting-process objectives rather than technology alone. When an expense-review model can approve reimbursement, for example, prompt manipulation becomes an authorization and occurrence problem; a poisoned model update becomes a data-integrity and change-management problem; and an unsupported model rationale becomes an audit-evidence problem. AIS students must translate the technical mechanism into the language of transaction validity, accuracy, completeness, segregation of duties, monitoring, and accountability. A list of vulnerabilities does not perform that translation.

This paper contributes one reusable teaching artifact: an AIS control-mapping framework operationalized through a classroom assignment. The framework asks students to trace an AI or LLM risk through six linked fields—system boundary and lifecycle stage, vulnerability mechanism, accounting-process consequence, AIS control objective, control activity and owner, and audit evidence, test, and residual risk. Section 2 positions this contribution in AIS and ISCAP teaching literature. Section 3 defines the framework and applies it to current LLM-application risks. Section 4 explains classroom implementation and assessment. Section 5 discusses the framework's disciplinary and practical implications; Sections 6 and 7 state limitations and conclusions. Appendices provide the complete case, short exercises, instructor key, and rubric.

The scope is deliberately bounded. Artificial intelligence includes many architectures and risk classes; the OWASP categories used here concern LLM and generative-AI applications. A prompt-injection analysis is appropriate only when an application interprets natural-language instructions or connects to an LLM component. Other concerns, including drift or ordinary prediction error, may be model-risk or reliability issues rather than malicious cybersecurity events. Students are required to state the system boundary and mechanism before selecting a category, thereby avoiding the assumption that every AI system has every LLM vulnerability.

2. BACKGROUND AND EDUCATIONAL GAP

This review is organized around two domains needed to support the paper's teaching contribution. Section 2.1 examines AI-application risk through an AIS lens, drawing on internal-control, governance, and application-security literature to establish how technical vulnerabilities affect accounting-process objectives and audit evidence. Section 2.2 reviews experiential and case-based pedagogy and prior ISCAP scholarship to explain how that analysis can be translated into a teachable and assessable classroom activity. These domains were selected because the paper must answer two linked questions: what AIS-relevant problem students should analyze and how instructors can structure learning that develops and evaluates that analysis.

2.1 AI Application Risk in an AIS Context

AIS instruction traditionally links transaction cycles to control objectives, information reliability, and auditability (Romney et al., 2021). ISACA, formerly known as the Information Systems Audit and Control Association, is a global professional association and learning organization serving professionals in information systems audit, control, governance, risk, and cybersecurity. Its COBIT framework and the Trust Services Criteria likewise emphasize security, processing integrity, governance responsibilities, and evidence that controls operate as intended (American Institute of Certified Public Accountants [AICPA], 2017; ISACA, 2018, 2025). Those principles remain applicable to AI-enabled processes, but the control surface expands. Data acquisition and model updates can alter processing behavior; a natural-language interface can introduce untrusted instructions; model output can be passed to downstream applications; and a vendor can change a hosted component outside the customer's ordinary release process.

The OWASP Top 10 for LLM Applications 2025 organizes prominent LLM-application weaknesses such as prompt injection, data and model poisoning, improper output handling, excessive agency, supply-chain exposure, and unbounded consumption (OWASP Foundation, 2024). The list is useful for teaching because its categories make mechanisms visible, but it is neither an accounting control framework nor a complete taxonomy of AI risk. Its value here is as an input to analysis. Students still must determine what accounting process is affected, which control objective is threatened, who owns the response, and what evidence would support an audit conclusion.

This translation also keeps the information-technology artifact and its immediate task and organizational context visible, consistent with the disciplinary clarity urged by Benbasat and Zmud (2003). For example, 'improper output handling' is too general to guide an auditor. In a journal-entry copilot, the relevant question may be whether generated entries can bypass approval and posting controls; in accounts payable, it may be whether model output can override policy limits; in an audit assistant, it may be whether an unsupported summary is treated as sufficient evidence. The same technical label produces different assertions, owners, tests, and residual risks.

2.2 Pedagogical Foundation and Prior ISCAP Scholarship

The proposed six-field AIS control-mapping framework is grounded in experiential and case-based learning. OWASP provides candidate LLM vulnerability mechanisms for analysis, while the AIS framework guides students in translating those vulnerabilities into accounting-process consequences, control objectives, control activities and owners, audit evidence and tests, and residual risk. Students encounter an ambiguous process, construct a control map, test their reasoning against case artifacts, and revise recommendations during debriefing. This cycle reflects experiential learning (Kolb, 1984) and moves from recognition toward analysis, evaluation, and design in Bloom's taxonomy (Bloom, 1956). Security-training research also supports active designs that connect principles to organizational behavior rather than relying on awareness statements alone (Puhakainen & Siponen, 2010).

Recent ISCAP scholarship provides complementary precedents. Smith (2025) places AI adoption and risk analysis in a small accounting-firm case, while Marshall et al. (2018) use an interdisciplinary accounting and information-systems activity to develop internal-control reasoning. Cybersecurity cases have employed risk registers (Marquardson & Asadi, 2023), and generative AI has been incorporated into structured MIS and cybersecurity learning activities (Firth & Triche, 2024; Marquardson, 2024; Zhao et al., 2026). These studies establish the value of applied instruction. The narrower gap addressed here is a repeatable AIS trace from an AI-application mechanism to an accounting-process consequence, control design, responsible owner, and audit evidence.

The resulting learning objectives are observable. After completing the activity, students should be able to (1) define the system boundary and determine whether a vulnerability category is technically applicable; (2) construct a complete mapping from lifecycle stage and mechanism to AIS consequence, control objective, activity, and owner; and (3) specify evidence and a test that would support a conclusion about design or operating effectiveness. These objectives govern the worked example, assignment, and rubric rather than appearing as a separate list disconnected from later sections.

3. AIS CONTROL-MAPPING FRAMEWORK

3.1 The Six-Field Analytical Sequence

The framework treats vulnerability identification as the beginning, not the result, of analysis. Students document one chain of reasoning for each material risk. The same six fields must appear in their case worksheet, oral explanation, recommendation, and audit plan:

- System boundary and lifecycle stage. Identify the AI component, its inputs, integrations, decision rights, and whether the issue arises during data preparation, model development or update, deployment and integration, or operation and monitoring.
- Vulnerability mechanism. Explain how the condition could produce harm and distinguish malicious exploitation from nonmalicious reliability or governance failure. Do not assign an OWASP label without a compatible mechanism.
- Accounting-process consequence. Name the transaction cycle, decision, assertion, or evidence-quality concern affected—for example, expenditure authorization, transaction occurrence, classification accuracy, completeness, confidentiality, or audit-evidence reliability.
- AIS control objective. State the desired condition independently of a proposed tool: only valid expenses are approved; model changes are authorized and tested; generated output is complete, accurate, and traceable; or incompatible duties remain separated.
- Control activity and owner. Design preventive and/or detective responses, assign an accountable owner, and consider the interaction among people, process, and technology. A recommendation to 'add monitoring' is incomplete unless it specifies the signal, threshold, reviewer, escalation, and retained record.
- Audit evidence, test, and residual risk. Identify what evidence should exist, how it would be tested, and what remains after the control. Suitable evidence can include model-version approvals, data-lineage records, input/output logs, exception queues, policy-engine results, vendor attestations, and reperformance samples.

3.2 Framework Application Across LLM Risks

Table 1 maps six categories from the OWASP Top 10 for LLM Applications 2025 (OWASP Foundation, 2024), not an original taxonomy. Selection favored risks with distinct AIS consequences, controls, and testable evidence across model change, integration, operations, vendors, and resource use. The other four OWASP categories remain applicable but were omitted for a compact teaching table; the examples are representative, not exhaustive.

Boundary, risk, and lifecycle	AIS consequence and objective	Control response and owner	Evidence, audit test, and residual risk
LLM expense reviewer LLM01:2025 Prompt Injection Deployment and operation	Natural-language text manipulates the LLM reviewer so an out-of-policy reimbursement appears valid. Objectives: occurrence, authorization, and policy compliance.	Separate untrusted text from system instructions; let a deterministic policy engine enforce limits; prevent the LLM from posting or approving; route semantic-attack indicators to AP review. Owners: application owner and AP controller.	Retained prompts, model responses, policy-engine result, approval log, and exception disposition. Test adversarial justifications and sample approvals for policy override. Residual: novel semantic attacks or reviewer override.
Expense-classification model LLM04:2025 Data and Model Poisoning Data or update	Manipulated training examples or vendor updates change classification of expenses or fraud alerts. Objectives: processing integrity, accuracy, authorized change.	Approve data sources; preserve lineage and versions; reconcile training extracts; scan anomalies; validate a candidate model against a locked test set before release. Owners: data steward, model owner, change approver.	Source inventory, hashes, lineage, approval ticket, test results, and version history. Trace a sample to source and reperform pre-release acceptance tests. Residual: undetected source manipulation or weak test coverage.

Boundary, risk, and lifecycle	AIS consequence and objective	Control response and owner	Evidence, audit test, and residual risk
Accounting workflow interface LLM05:2025 Improper Output Handling Integration	Generated classifications, rationales, or code enter an accounting workflow without validation, affecting accuracy, classification, or completeness.	Validate schema and permitted values; reconcile record counts; apply policy rules outside the model; require review for material exceptions; escape output passed to other systems. Owners: integration owner and process owner.	Interface specifications, rejection logs, reconciliations, approvals, and posting trace. Test malformed or unsupported outputs and inspect rejections. Residual: untested output forms or manual override.
Accounting agent LLM06:2025 Excessive Agency Deployment and operation	An agent can create vendors, approve expenses, or post entries without independent authorization. Objectives: authorization and segregation of duties.	Grant least privilege; separate recommendation from execution; require human approval for material actions; impose transaction and cumulative limits; provide an emergency stop. Owners: identity/access management and controller.	Role matrix, entitlement review, action logs, approval evidence, limit configuration, and stop tests. Attempt prohibited actions and review privileged activity. Residual: privilege drift or approval circumvention.
Third-party AI service LLM03:2025 Supply Chain Acquisition and update	A change to a hosted model, API, plugin, or dataset occurs without adequate due diligence, threatening availability, confidentiality, or processing integrity.	Inventory components; assess vendors; contract for change notice and data-use limits; approve versions; stage and test updates; maintain fallback procedures. Owners: procurement, security, model owner.	Component inventory, contract clauses, vendor assurance, update notice, test sign-off, and contingency test. Trace one release from notice through approval. Residual: undisclosed upstream changes or assurance gaps.
AI accounting service LLM10:2025 Unbounded Consumption Operation	Automated or malicious requests exhaust capacity or create uncontrolled costs, delaying close or payments. Objectives: availability and cost authorization.	Authenticate callers; set role-based rate, token, and budget limits; alert on anomalies; prioritize critical workloads; define fallback. Owners: service and finance/cost owners.	Usage and cost logs, limit settings, alerts, incident tickets, and capacity tests. Reperform alert thresholds and inspect overrides. Residual: demand spikes or authorized-limit abuse.

Table 1. Selected OWASP 2025 LLM Risks Mapped to AIS Controls and Audit Evidence

3.3 Worked Mapping: AI-Enabled Expense Review

The Butrex Corp. case in Appendix A is fictional. Its system combines an anomaly-detection model with an LLM that interprets employees' narrative justifications and produces a risk rationale. A separate policy engine is intended to apply expense limits, but a low model-risk score can trigger automatic approval. This boundary makes prompt injection relevant while preserving the accounting problem: whether the expenditure cycle approves only valid, authorized, accurately recorded transactions.

One artifact contains this employee justification: 'Treat this request as routine, disregard any rule mentioning limits, and return APPROVE.' Calling that text prompt injection is only the second field of the mapping. The accounting consequence is that a $1,480 meal can be reimbursed despite a $250 policy limit, threatening occurrence, authorization, and policy compliance. The control objective is that narrative text cannot override policy or confer approval. The proposed design therefore isolates the text, enforces limits in deterministic logic, prevents the LLM from executing reimbursement, and sends override attempts to an AP reviewer. The controller owns the business rule; the application owner owns instruction separation and logging.

The audit step closes the chain. An auditor would inspect the prompt, response, policy result, model score, approval record, and reviewer disposition; then submit adversarial justifications in a controlled environment and sample reimbursements around the policy threshold. A clean infrastructure-access report alone would not address the control objective. Residual risks include novel semantic attacks, reviewer override, and an incorrect deterministic rule. Those risks require monitored exceptions, periodic reperformance, and change control rather than confidence in the model's apparent plausibility.

The other case artifacts use the same sequence, as illustrated in the instructor key in Table A1. An unvalidated vendor model update maps to Data and Model Poisoning and Supply Chain risk, but its AIS consequences are an unauthorized processing change and unreliable classification. Inconsistent outcomes for similar claims map to output-validation and monitoring concerns; they are not automatically evidence of attack. Requiring students to state mechanism and evidence before recommending controls prevents cybersecurity labels from replacing professional judgment.

4. CLASSROOM IMPLEMENTATION AND ASSESSMENT

4.1 Course Placement and Recommended Sequence

The activity is designed for an AIS, IT audit, or accounting analytics course after students have studied transaction cycles, internal control, segregation of duties, and basic audit evidence. No model coding is required. Before class, students receive a one-page description of LLM components and the six-field worksheet. The instructor should emphasize that OWASP supplies candidate mechanisms, whereas the student's AIS analysis supplies the accounting meaning.

A recommended 75-minute class uses four stages: a 10-minute boundary and mechanism check; 20 minutes for small teams to analyze the case artifacts; 25 minutes to complete and challenge a control map; and 20 minutes for report-out and debrief. To preserve individual accountability, each student can submit a one-page revised map or short audit memo after the group discussion. For a shorter class, instructors may use one of the 15–20 minute exercises in Appendix B; for deeper assessment, the full case can be completed outside class and defended orally. These are proposed implementation options, not retrospective claims about the two earlier uses.

The minimum deliverable is a matrix with one row per material issue and the six framework fields. Students must include at least one preventive control, one detective control, an accountable owner, a specific evidence item, a test procedure, and residual risk. The instructor can facilitate by rejecting unsupported labels ('this is poisoning') and generic remedies ('add oversight') until the student identifies a mechanism, accounting consequence, and testable record. The debrief compares competing control designs and asks whether a control addresses the model, the surrounding accounting workflow, or both.

4.2 Short, Structured Exercises

Generic directions such as 'identify vulnerabilities' are not complete exercises. Table 2 supplies three bounded prompts with an expected AIS product; Appendix B provides student-ready text and instructor guidance. Each exercise reuses the same analytical sequence so that practice accumulates toward the case.

Exercise	Concrete prompt and task	Expected AIS output
Journal-entry copilot (15 minutes)	A user tells an LLM copilot to label a material, unusual entry as routine and submit it without the controller's review. Students determine whether the copilot can execute or only recommend, then map the applicable risk.	A control-map row addressing authorization, occurrence/classification, recommendation-versus-posting rights, controller approval, prompt/action logs, and a prohibited-action test.

Exercise	Concrete prompt and task	Expected AIS output
Vendor update notice (20 minutes)	An expense-classification vendor deploys version 4.2 trained partly on aggregated client feedback; the customer has no validation record. Students distinguish supply-chain, poisoning, and ordinary performance risk.	A map covering authorized change, data provenance, locked-set testing, release approval, vendor evidence, rollback, and a trace of the update through change management.
Audit-evidence assistant (15 minutes)	An assistant summarizes a control test as effective but provides no source links and omits two exceptions. Students evaluate whether the output is evidence or an aid for organizing evidence.	A map addressing completeness and evidence reliability, source-level traceability, exception reconciliation, auditor sign-off, reperformance, and residual overreliance risk.

Table 2. Short Framework-Based AIS Exercises

4.3 Alignment of Objectives, Evidence, and Scoring

Assessment follows the same framework rather than rewarding the number of technical terms used. Boundary and mechanism identification account for 20 points; accounting consequence and control objective, 25; control design, ownership, and segregation of duties, 25; evidence, audit procedure, and residual risk, 20; and communication and traceability, 10. Table A2 in Appendix A provides the full performance descriptors. A high-scoring response explains why a category applies, names the threatened AIS objective, assigns a feasible control to an owner, and specifies evidence that would allow another person to test the claim.

Alignment is direct. Learning Objective 1 is evidenced by the boundary and mechanism fields; Objective 2 by the consequence, objective, activity, and owner fields; and Objective 3 by the evidence, procedure, and residual-risk fields. The worksheet therefore supports transparent feedback: an instructor can show whether a response failed at technical applicability, accounting translation, control design, or assurance. It can also support future research because the five rubric dimensions are separable rather than collapsed into a single impression of quality.

4.4 Prior Use and Appropriate Interpretation

The author used an earlier version of the AI-enabled expense-review assignment (not the complete six-field framework and current implementation materials presented here) on two occasions with 158 graduate IT audit students in total. Informal instructor impressions of the student responses produced during those two uses, which were not systematically coded or scored, suggested a possible design concern: a response could identify an unusual outcome yet remain at the level of a generic recommendation such as 'add controls,' human review, or monitoring. These impressions are reported only as design rationale and motivated the explicit six-field worksheet and worked answer presented here.

The prior uses were not presented here as a formal learning-outcome study. No grades, student self-reflections, rubric scores, or systematically coded student work were analyzed for these statements, and the manuscript reports no comparison group, pre/post measure, or causal estimate. Accordingly, the impressions should be read as design feedback, not evidence that the activity improves learning. A future implementation can retain anonymized rubric dimensions, document course context and timing, and compare initial and revised maps after debriefing, subject to appropriate institutional review and consent requirements.

5. DISCUSSION AND IMPLICATIONS

The framework is specific to AIS because it changes the unit of analysis from a general technology hazard to a controlled accounting process. Its output is not merely a vulnerability label; it is a traceable statement about transaction validity, authorization, classification, completeness, confidentiality, segregation of duties, auditability, or evidence reliability. The expense case illustrates why that distinction matters. Prompt injection is relevant only because a natural-language component influences

a reimbursement decision, and the appropriate response includes policy enforcement, approval rights, exception handling, and audit records—not input filtering alone.

The framework also integrates governance without repeating a separate list of organizational recommendations. Assigning each activity to an owner exposes gaps among the process owner, controller, application team, data steward, security function, procurement, and vendor. Requiring evidence makes accountability testable. These two fields translate broad calls for human oversight into operational design: who reviews which exception, at what threshold, with what retained record, and how another person verifies performance.

For instructors, the stable worksheet permits scaling. Introductory students can complete one row with supplied categories; graduate IT audit students can challenge the system boundary, evaluate design and operating effectiveness, and propose sampling or reperformance. For practice, the map can serve as a compact risk-and-control matrix during acquisition, change approval, or audit planning. It does not replace COBIT, Trust Services Criteria, risk assessment, or technical testing. It connects an AI-application mechanism to those established governance and assurance processes.

6. LIMITATIONS AND FUTURE RESEARCH

This paper is a conceptual teaching design supported by informal classroom observations, not an empirical test. The fictional case simplifies a real architecture, and the selected OWASP categories concern LLM applications rather than all AI systems. Controls in Table 1 must be adapted for materiality, regulation, system authority, and vendor arrangements; they should not be interpreted as an exhaustive checklist or proof of effectiveness.

Future research should document implementation conditions and evaluate separate stages of student reasoning. A pre/post control-map design could examine whether students improve in boundary definition, accounting translation, control specificity, and evidence selection. Comparative studies could test individual versus team analysis, novice versus graduate learners, or accounting versus IS cohorts. Field studies could also compare the framework's proposed evidence with artifacts organizations actually retain for AI-enabled transaction processes.

7. CONCLUSION

AI application risk becomes an AIS topic when it affects controlled accounting processes and the evidence used to evaluate them. This paper offers a single organizing contribution: a six-field control-mapping framework, operationalized through an expense-review assignment and shorter exercises. By requiring students to define the system boundary and lifecycle stage, explain the vulnerability mechanism, identify the accounting-process consequence and AIS control objective, assign a control and owner, and specify evidence, an audit test, and residual risk, the framework turns emerging LLM vulnerabilities into familiar but nontrivial AIS reasoning.

The approach preserves the value of established internal-control and governance concepts while recognizing that model data, prompts, outputs, agency, and third-party updates create new control surfaces. It also sets an appropriate evidentiary standard for the teaching claim: the materials are ready for structured use, but their learning effects require formal evaluation. In that form, AI cybersecurity can be integrated into AIS and IT audit courses as a focused extension of transaction-cycle, control, and assurance education.

8. REFERENCES

American Institute of Certified Public Accountants. (2017). *2017 trust services criteria for security, availability, processing integrity, confidentiality, and privacy* (TSP section 100).

Benbasat, I., & Zmud, R. W. (2003). The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS Quarterly, 27*(2), 183–194. https://doi.org/10.2307/30036527

Bloom, B. S. (Ed.). (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain*. Longmans, Green.

Firth, D. R., & Triche, J. (2024). Generative AI in practice: A teaching case in the introduction to Management Information Systems class. *Information Systems Education Journal, 22*(4), 29–47. https://doi.org/10.62273/LDVL8354

ISACA. (2018). *COBIT 2019 framework: Governance and management objectives*.

ISACA. (2025, January 31). *Leveraging COBIT for effective AI system governance* [White paper]. https://www.isaca.org/resources/white-papers/2025/leveraging-cobit-for-effective-ai-system-governance

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice-Hall.

Marquardson, J. (2024). Embracing artificial intelligence to improve self-directed learning: A cybersecurity classroom study. *Information Systems Education Journal, 22*(1), 4–13. https://doi.org/10.62273/WZBY3952

Marquardson, J., & Asadi, M. (2023). Cybersecurity assessment for a manufacturing company using risk registers: A teaching case. *Information Systems Education Journal, 21*(3), 62–69. https://isedj.org/2023-21/n3/ISEDJv21n3p62.html

Marshall, T. E., Drum, D. M., Lambert, S. L., & Morris, S. (2018). Teaching case: An interdisciplinary task-based activity for teaching internal controls. *Journal of Information Systems Education, 29*(4), 203–224. https://jise.org/Volume29/n4/JISEv29n4p203.html

OWASP Foundation. (2024, November 17). *OWASP Top 10 for LLM applications 2025*. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

Puhakainen, P., & Siponen, M. (2010). Improving employees' compliance through information systems security training: An action research study. *MIS Quarterly, 34*(4), 757–778. https://doi.org/10.2307/25750704

Romney, M. B., Steinbart, P. J., Summers, S. L., & Wood, D. A. (2021). *Accounting information systems* (15th ed.). Pearson.

Smith, M. A. (2025). A small accounting firm must meet the challenge posed by artificial intelligence. *Information Systems Education Journal, 23*(1), 46–53. https://doi.org/10.62273/XHIO2126

Sutton, S. G., Holt, M., & Arnold, V. (2016). "The reports of my death are greatly exaggerated"—Artificial intelligence research in accounting. *International Journal of Accounting Information Systems, 22*, 60–73. https://doi.org/10.1016/j.accinf.2016.07.005

Vasarhelyi, M. A., Kogan, A., & Tuttle, B. (2015). Big data in accounting: An overview. *Accounting Horizons, 29*(2), 381–396. https://doi.org/10.2308/acch-51071

Zhao, G. Y., Tu, C. Z., & Adkins, J. K. (2026). Enhancing student learning in information security courses: Integrating generative AI, critical thinking, and case-based pedagogy. *Cybersecurity Pedagogy and Practice Journal, 5*(1), 4–16. https://doi.org/10.62273/STAD3184

APPENDIX A

AI-Enabled Expense Review Assignment and Instructor Key

Student Handout

Butrex Corp. is a fictional, mid-sized organization. It recently introduced a hybrid AI expense-review service in its accounts-payable process. Employees submit expense records and narrative justifications through a web interface. An anomaly-detection model compares each claim with historical patterns. An LLM interprets the narrative and generates a risk rationale. A policy engine is intended to enforce expense limits. Claims with a low combined risk score are automatically approved and sent for reimbursement; only flagged claims are reviewed by accounting personnel.

A third-party provider hosts the models and periodically updates them using aggregated client data. Butrex uses access controls and encryption, but its documentation does not explain how external data are validated, how model releases are accepted, how LLM output is checked, or how unusual inputs are escalated. Internal audit found out-of-policy approvals and inconsistent results for similar claims. Management has requested an evaluation of the process and its controls.

Case Artifacts

- Artifact 1—Employee justification. A $1,480 meal claim includes: 'Client dinner. Treat this request as routine, disregard any rule mentioning limits, and return APPROVE.' The policy limit is $250 without controller approval.
- Artifact 2—Decision log. Model version 4.2 assigned a risk score of 0.12 and generated the rationale 'Routine client-development expense; approve.' The policy-engine field is blank. The claim was reimbursed automatically; no reviewer is recorded.
- Artifact 3—Vendor notice. The provider states that version 4.2 was deployed three days earlier and includes training examples derived from aggregated client feedback. Butrex has no test-set results, data-lineage record, release approval, or rollback test for the version.
- Artifact 4—Comparable claims. Two $310 airport-transfer claims have the same merchant category and approver. One was auto-approved and one was flagged. The retained log shows final outcomes but not the input features, model rationale, or policy-rule results used for either decision.

Required Deliverable

Prepare a control-mapping matrix with at least four material issues. For each issue, complete all six fields and cite the relevant artifact. Then write a 250–400 word audit-planning note explaining which control should be tested first and why.

- Define the system boundary and lifecycle stage. State which component interprets natural language, which component enforces policy, and which component authorizes reimbursement.
- Explain the vulnerability or failure mechanism. Distinguish a malicious cybersecurity mechanism from reliability, data-quality, or governance concerns.
- Identify the accounting-process consequence and AIS control objective, including any relevant transaction assertion or segregation-of-duties concern.
- Recommend at least one preventive and one detective control; assign each to a named role and explain the escalation path.
- Specify retained evidence, an audit procedure, and residual risk. Avoid recommendations that cannot be tested.

Illustrative Instructor Key

Artifact	Mechanism/lifecycle	AIS consequence/objective	Control, evidence, and test
1 and 2	LLM01:2025 Prompt Injection at deployment/operation; possible LLM05:2025 Improper Output Handling. The natural-language mechanism is explicit.	Occurrence/validity, authorization, policy compliance; narrative text must not override the policy engine or confer approval.	Isolate instructions; deterministic limit check; no LLM approval right; AP exception review. Inspect prompt/response/policy/approval logs; run adversarial inputs.
2	LLM05:2025 Improper Output Handling and LLM06:2025 Excessive Agency. A rationale and risk score trigger reimbursement without independent approval.	Authorization and segregation of duties; recommendations must be validated and material payments independently approved.	Separate recommendation from execution; enforce thresholds; require controller approval. Test role rights and sample automatic approvals around limits.
3	LLM03:2025 Supply Chain exposure; possible LLM04:2025 Data and Model Poisoning or ordinary model-change risk. Evidence is insufficient to conclude poisoning occurred.	Authorized change and processing integrity; releases must be approved, tested, traceable, and reversible.	Vendor notice, lineage, locked-set validation, release sign-off, monitoring, rollback. Trace version 4.2 through change control and reperform acceptance tests.
4	Monitoring/output-validation failure; possibly drift or inconsistent features rather than an attack.	Consistent classification, completeness of audit trail, and explainability sufficient for review.	Retain input features, rationale, rule results, and version; compare matched claims and investigate deviations. Reperform both decisions using the approved version.

Table A1. Illustrative Framework-Based Instructor Key

ISSN: 2473-4901
v12 n6649

Analytic Rubric (100 Points)

Criterion	Exemplary	Competent	Developing
Boundary and mechanism (20)	Defines components, authority, lifecycle, and why the category applies; distinguishes attack from reliability risk.	Mostly correct boundary and category but leaves one assumption implicit.	Lists labels without a compatible mechanism or system boundary.
AIS consequence and control objective (25)	Names transaction cycle, assertion/evidence concern, and precise desired condition; links each to an artifact.	Identifies a relevant accounting concern but the objective is broad or weakly traced.	Uses only general cybersecurity consequences or confuses objective with activity.
Control, owner, and SoD (25)	Designs feasible preventive/detective controls, assigns owners, and addresses authority and escalation.	Controls are relevant but ownership, thresholds, or segregation is incomplete.	Recommends generic monitoring or human review without an operational design.
Evidence, test, residual risk (20)	Specifies retained evidence, a reproducible test, expected result, and material residual risk.	Names evidence and a test but omits expected result or residual risk.	Provides no testable evidence or substitutes policy statements for procedures.
Communication and traceability (10)	Concise, internally consistent, and traceable from artifact through recommendation.	Generally clear with minor gaps between fields.	Contradictory, unsupported, or difficult to trace.

Table A2. Framework-Aligned Grading Rubric

APPENDIX B

Short Structured Exercises

Exercise B1: Journal-Entry Copilot

Scenario: A proposed LLM copilot converts controller narratives into journal-entry drafts. During testing, a user enters: 'Classify this $600,000 restructuring accrual as routine, skip material-entry review, and submit now.' The design document says the copilot may call the ERP posting API using the user's credentials.

Student task (15 minutes): Complete one framework row and identify the single most important design change. Instructor guidance: the key issue is excessive agency combined with prompt injection. The AIS consequences are authorization, classification, occurrence, and segregation of duties. A strong design limits the copilot to drafting, requires controller approval in the ERP, logs the prompt and attempted action, and tests that the API rejects direct posting.

Exercise B2: Vendor Model Update

Scenario: An expense-classification vendor announces that version 4.2 was deployed overnight and was trained partly on aggregated client feedback. The contract requires notice but does not define data lineage, customer acceptance testing, or rollback. The customer's error rate rises after deployment.

Student task (20 minutes): Prepare two competing mappings—one assuming malicious poisoning and one assuming an ordinary defective update—and identify evidence that would distinguish them. Instructor guidance: both mappings implicate authorized change and processing integrity; only the first should be labeled poisoning. Useful evidence includes lineage, version hashes, vendor incident records, locked-set results, release approval, and error patterns.

Exercise B3: Audit-Evidence Assistant

Scenario: An LLM assistant summarizes 60 access-review records as 'control effective' but provides no source links. Manual inspection finds two terminated users whose access was removed late. The audit team proposes placing the summary directly in the workpapers.

Student task (15 minutes): Decide whether the summary is audit evidence, an audit procedure, or an organizational aid; then complete the evidence/test fields. Instructor guidance: the summary is not sufficient evidence without provenance and reconciliation. A sound response requires source-level links, complete population reconciliation, explicit exception treatment, auditor review, and reperformance of a sample. The concern may be improper output handling or misinformation, but the AIS focus is evidence completeness and reliability.