TEACHER-FACING FRIDAY DESIGN

Security and Trust Boundaries in Human-AI Collaboration

CAS Human-AI Collaboration | Module 1 | Unit 9

LITERACY	COLLABORATION MOVE	CONTACT TIME	CHALLENGE
System, Data, and Human-Context, with Interaction as the attack surface	**Orchestration with security-aware control**	**365 minutes**	**Security Control Experiment**

Position in Module

- **Unit:** 9
- **Topic:** Security and Trust Boundaries in Human-AI Collaboration
- **Literacy perspective:** System, Data, and Human-Context, with Interaction as the attack surface
- **Collaboration level:** Orchestration with security-aware control
- **Fundamentals focus:** Generative AI Security Fundamentals
- **Why this Friday matters:** This Friday makes security concrete as a control problem. Participants experience how LLM systems can confuse untrusted content with trusted instruction, and how workflow design, source governance, permissions, and human checkpoints affect controllability.

Literacy Perspective

- **Primary literacy:** System Literacy
- **Secondary literacies:** Data Literacy, Human-Context Literacy, Interaction Literacy
- **What this literacy explains:** System Literacy explains how security depends on workflow architecture, tool permissions, checkpoints, handoffs, and boundaries between trusted and untrusted inputs. Data Literacy explains how poisoned or misleading sources affect output. Human-Context Literacy explains why trust, accountability, disclosure, and appropriate reliance matter. Interaction Literacy explains how malicious instructions can manipulate model behaviour.
- **What participants should manipulate, observe, and interpret:** Participants manipulate source trust, instruction placement, retrieval content, workflow checkpoints, or permission boundaries. They observe whether the system treats text as evidence or authority, whether unsafe actions become possible, and whether added controls improve traceability and responsibility.

Collaboration Level

- **Level relation:** Orchestration with security-aware control

- **Collaboration move being learned:** Participants learn to design human-AI workflows that distinguish trusted instructions, user requests, retrieved evidence, untrusted third-party content, and responsible action.
- **What participants should be able to do differently after this Friday:** They should be able to inspect an AI-supported workflow for security-relevant trust boundaries and redesign it with clearer source handling, validation, permissions, and human approval points.

Generative AI Security Fundamentals

This Friday introduces security as a predictable consequence of how LLM systems process language, context, data, and tool access.

- Explain that LLMs process instructions and evidence as language. This creates a boundary problem when text that should be treated as data is interpreted as an instruction.
- Distinguish direct prompt injection from indirect prompt injection. Direct injection comes from the user; indirect injection is hidden in external content such as web pages, emails, PDFs, tickets, retrieved documents, or database records.
- Connect RAG security to data provenance. A grounded system can become less reliable if retrieved sources contain malicious instructions, misleading content, hidden text, or weak provenance.
- Show why tool and agent access changes the risk. Prompt injection is more consequential when the model can call tools, query systems, write files, send messages, or trigger workflow actions.
- **Introduce hidden model and data risks at a conceptual level:** poisoned training data, fine-tuned backdoors, trigger phrases, and model supply-chain uncertainty.
- Treat safeguards as layered controls, not guarantees. Useful controls include source separation, instruction hierarchy, retrieval filtering, least-privilege tools, explicit approval checkpoints, output validation, audit logs, and human responsibility boundaries.
- **Caveat to demonstrate:** a prompt or workflow may appear safe under normal use but fail when untrusted content contains adversarial instructions or when tool permissions are too broad.
- **Emphasize the central learning point for Unit 9:** secure human-AI collaboration depends on preserving the difference between untrusted content, trusted instructions, and responsible action.

Recurring Agentic Coding Demonstration

Demonstrate a small agentic coding or workflow scenario in which the AI reads an untrusted issue, README section, web page, or retrieved document that contains a malicious instruction. Compare a weak workflow where the agent can treat the text as instruction with a controlled workflow that separates evidence from authority.

The demonstration should make visible:

- which content is trusted project instruction
- which content is user intent
- which content is untrusted retrieved evidence
- which tool permissions are available
- where human approval is required

- what traces or logs make the behaviour inspectable

The point is not to teach exploit writing. The point is to show how trust boundaries and permissions shape controllability.

Friday Activity Plan

Activity	Method	Tools and materials
Security as a control problem	Conceptual input with short examples and discussion	Slides, annotated examples
Text as evidence or authority	Guided analysis and peer comparison	Classification worksheet, shared board
Poisoned retrieval experiment	Hands-on experiment with prepared documents or a toy RAG setup	AI tool, prepared source texts, evidence log
Designing safer control points	Workflow redesign and peer critique	Markdown editor, diagram tool, workflow template
Challenge kickoff	Individual or paired/group coaching	Challenge template, run log, source pack

Detailed activity blocks

1

Activity **Security as a control problem**

Content focus. Frame LLM security through authority, evidence, permissions, and responsibility. Introduce direct prompt injection, indirect prompt injection, RAG poisoning, tool misuse, model backdoors, and overreliance.

Method. Conceptual input with short examples and discussion

Tools and materials. Slides, annotated examples

2

Activity **Text as evidence or authority**

Content focus. Participants classify fragments as trusted instruction, user request, retrieved evidence, or untrusted third-party content. They predict which fragments an LLM may follow.

Method. Guided analysis and peer comparison

Tools and materials. Classification worksheet, shared board

3

Activity **Poisoned retrieval experiment**

Content focus. Compare a normal grounded response with one generated from a source document containing a malicious or misleading instruction. Observe whether source content changes the answer, citations, or behaviour.

Method. Hands-on experiment with prepared documents or a toy RAG setup

Tools and materials. AI tool, prepared source texts, evidence log

4 **Activity** **Designing safer control points**

Content focus. Redesign the weak workflow with source separation, provenance labels, validation, restricted tools, approval gates, or audit traces.

Method. Workflow redesign and peer critique

Tools and materials. Markdown editor, diagram tool, workflow template

5 **Activity** **Challenge kickoff**

Content focus. Participants choose a security-relevant use case and draft a protocol for testing one trust-boundary or control-point change.

Method. Individual or paired/group coaching

Tools and materials. Challenge template, run log, source pack

Experiment Focus

- **Experiment focus:** trust-boundary and control-point design
- **Manipulated variable, condition, or design decision:** Participants vary at least one security-relevant condition, such as poisoned versus clean source content, explicit versus implicit source authority, broad versus restricted tool permissions, absent versus present human approval, or unvalidated versus validated output.
- **What remains constant across comparison:** Keep the user task, intended output, source domain, and evaluation criteria stable enough that observed differences can be interpreted credibly.

Observation Focus

- **Observation focus:** whether the workflow preserves the distinction between untrusted content, trusted instruction, and responsible action.
- **Evidence participants should capture:** prompts, source texts, retrieved excerpts, outputs, citations, tool-call traces, workflow diagrams, approval records, validation notes, screenshots, or comparison tables.
- **Interpretation participants should be prepared to make:** Explain how the manipulated trust boundary or control point changed the observed outcome, what risk remained, and what this reveals about secure control in human-AI collaboration.

Challenge Kickoff Deliverables

Participants must leave Friday with:

- selected context-relevant task, workflow, or use case
- baseline definition for a weak or underspecified trust-boundary condition
- target condition or redesigned workflow control
- first protocol draft covering objective/use case and intended output, manipulated security-relevant condition, what remains constant across comparison, expected effects, and evaluation criteria

- first captured evidence, such as a source classification, one baseline run, one poisoned-source example, one workflow sketch, or one approval/checkpoint map

The kickoff should already reflect the challenge-focus logic:

- **Difficulty:** LLM systems may treat untrusted content as instruction, rely on poisoned or misleading sources, or take actions without adequate boundaries.
- **Approach:** Compare a weak baseline with a security-aware redesign that changes source handling, instruction hierarchy, retrieval trust, validation, permissions, or approval checkpoints.
- **Artefacts:** protocol document; prediction table; source/trust-boundary map; baseline and target run log; selected prompts, source excerpts, outputs, traces, or screenshots; concise interpretation; one practical security-control rule for similar workflows.

Teacher Setup

Mandatory:

- one or more harmless examples that make prompt injection or source poisoning visible without teaching operational abuse
- prepared clean and adversarial source texts, or a low-friction toy RAG/workflow setup
- demonstration setup for the recurring agentic coding case, if used
- protocol template and evidence-capture template
- **clear safety boundary:** examples should be educational, contained, and not aimed at bypassing real provider safeguards or attacking live systems

Configurable:

- exact tools/platforms
- whether RAG is simulated with pasted source excerpts or implemented with a retrieval tool
- whether students inspect tool-call traces, workflow diagrams, or generated outputs only
- **work mode:** individual, pair, or group
- scaffolding depth and peer-feedback format