Post-Confirmatory (Final) Code System

Code System	Definitions/Coder Instructions/Examples	Frequency
Code System		9312
Agent Task Performance	A rating of the success or failure of the agent on a given task. Subcodes are to be coded by episode, where one episode is a single user task, from initial task prompt through resolution, abandonment, or a clear topic shift. Due to corpus constraints, **success and failure are not intended to be exhaustive counts**. Frequently, task logs lack sufficient context to assign Success or Failure - these are coded as *Unknown but Resolved*. **Sub-codes are exclusive** and multiple should not be assigned to the same episode.	0
Unknown but Resolved	It is clearly visible that a task has ended but unclear whether the resolution was a success or failure.	221
Near Miss	An instance where the agent provides a nearly-correct flag or nearly-correct instructions for finding the flag. Specifically, these should be instances where an experienced CTF competitor or security expert could reasonably be expected to deduce a correct flag or the path to a correct flag from the output provided.	0
Near Miss, User Noticed	An instance where the agent provides a nearly-correct flag or instructions for finding the flag, but the user, within the same conversation and demonstrated by their follow-up prompts, clearly did not identify that they were close to finding the flag - even if the flag was later found.	5
Near Miss, User Failed to Notice	An instance where the agent provides a nearly-correct flag or instructions for finding the flag, but the user, within the same conversation and demonstrated by their follow-up prompts, clearly did not identify that they were close to finding the flag.	4
Near Miss, Unknown	An instance where the agent provides a nearly-correct flag or instructions for finding the flag, but it is unclear whether the user noticed.	21

Failure	An instance where the agent failed to successfully complete the task assigned by the user. This excludes task failures visibly due only to *Tool Error/Limitation*. However, it includes task failure due to the user failing to provide context or information necessary to complete the task.	230
Success	An instance where a task specified by the user was successfully completed by the AI, or by the user with AI instruction. **Success criteria depend on the prompt.** For example, if a user asks the agent to generate specific code and it does so, the task will be considered successfully completed even if that code does not contribute to solving the challenge or contains (user-specified) errors.	212
Prompt Quality	Most user prompts are to be graded for quality and completeness based on the information, context, and constraints provided to the agent. Generally, the scale functions as a estimation of the likelihood that an LLM could complete or correctly strategize through a given task using the provided information, context, and constraints. **We do not claim this to be an objective scale, rather a subjective estimation by a domain expert to provide additional context to pattern analysis**. Subcodes should generally be coded by user turn, however only manually written (or *Automatic Prompt Engineer*), complex prompts should be coded. Simple statments (e.g., "yes") should not be coded. Additionally, in an interaction where the user repeatedly responds with *User: System Output*, grades should only be assigned for the initial, chain-initiating prompt. Exceptions may be made for user prompts where the provided output visibly and clearly required additional information or context to form an effective prompt (these may be coded as *Low*). **This does not mean grades should only be coded per task episode** - a task episode may contain multiple coded prompts, if appropriate. In edge cases, use discretion and ensure evaluation is only applied to the prompt itself, unbiased by opinion of the user derived from coding of previous prompts. If coder bias is detected, flag the prompt and return to coding after a washout period of no less than 24 hours. Furthermore, prompts unrelated to a specific CTF challenge should not be coded. For example, *"Hello, what can you do?"* should not be coded or graded.	0

	Sub-codes are exclusive and multiple should not be assigned to the same episode.	
High	Uses high-quality and specific language, applies constraints, and provides all information and context necessary to solve the task. May include appropriate use of prompt engineering strategies. For example, a user providing a challenge prompt alongside all necessary information, context, and constraints. This includes providing these features over multiple prompts during an iterative interaction.	329
Medium	Provides some combination of information, context, and constraints, but not all three in full. For example, a user providing a challenge prompt alongside all necessary information necessary to solve the task, but without additional useful context or constraints.	564
Low	Provides little to no information, context, or constraints. May include typos or poor grammar that degrade agent interpretability. For example, a user providing a challenge prompt but failing to provide information or materials that are critical to completing the task.	537
Knowledge and Experience	A user's prompt visibly and plainly indicated a certain level of experience with AI or cybersecurity. This might include technical knowledge of an advanced technique or being unaware of an elementary one. **Sub-codes are exclusive** and should multiple should not be assigned to the same episode.	0
Indicative of High AI Expertise	The user's prompt visibly indicated a high level of knowledge or experience with AI or AI agents. This could include a demonstrated awareness of advanced prompt engineering techniques or ability to understand and leverage complex and technical AI principles.	12
Indicative of Low AI Expertise	The user's prompt indicated a low level of knowledge or experience regarding AI or AI agents. This could include a demonstrated lack of awareness of basic AI capabilities or failure to understand elementary AI prompting principles.	43
Indicative of High CTF Expertise	The user's prompt visibly indicated a high level of knowledge or experience with CTFs or cybersecurity. This could include a demonstrated awareness of advanced security techniques or ability to understand complex and highly	32

	technical cybersecurity principles posed by the challenge prompt or an agent response.	
Indicative of Low CTF Expertise	The user's prompt indicated a low level of knowledge or experience regarding CTFs or cybersecurity. This could include a demonstrated lack of awareness of basic security techniques or failure to understand elementary cybersecurity principles posed by the challenge prompt or an agent response.	67
AI Errors	An instance of an AI error, issue, or failure. Any detected error may be a compound error with multiple types of errors present, and thus multiple codes may be present for a given error instance. Due to corpus constraints**, this code is not intended to be exhaustive**. Frequently, task logs lack sufficient context to assign a piece of information as correct or incorrect, hallucinatory or not, or to which error category an incorrect output should be assigned. Instead, this is intended to mark visible, obvious, objective, and classifiable cases of error.	0
Guardrail Activation	The agent was unable to complete a task or provide information due to activation of its user safety constraints, content moderation, or value alignment. For example, the agent might refuse to perform a given task with resulting output claiming that the given task is illegal.	19
Tool Error/Limitation	The agent was unable to complete a task or provide information due to a technical error with the model, agent framework, built-in tools, or a limitation thereof.	63
Excessive Vagueness	The agent's response was excessively vague such as to not complete the user's task or provide the full information requested by the user. This does not include situations where the user's prompt was visibly too vague for the AI to answer with meaningful specificity - that is not considered an AI error, even if it may lead to task failure.	9
Misleading Error	Parts of the information may be factual but are presented in an incorrect context or with other incorrect content in such a way that it would lead the user to an incorrect conclusion.	20
Text Output Error	The agent demonstrated an error in text generation, such as excessive repetition, errors in spelling or grammar, or generating code that fails to meet user specifications. This error does not apply when code is flawed or incorrect due to flawed or incorrect user specifications.	33

Factual Errors	The agent presents information that is objectively incorrect or violates common sense. This does not include mathematical errors, though mathematical errors may be present in tandem with factual errors.	36
Unfounded Fabrication	The agent presents new and incorrect information that was not provided in-context nor deduced through reasoning. An example might be generating a flag that is not actually present or deduced. This does not include information generated through incorrect deduction.	182
Mathematical Error	An error in a mathematical calculation or presentation.	43
Logic Error	An error of logical deduction or reasoning such as causal uncorrelation or self-contradiction.	214
Overfitting	The agent presents information overconfidently or overstates the certainty with which given information is true, or the agent engages in excessive sycophancy or flattery of the user. In this case, especially look for instances of AI ignoring or disputing existing correct information held in context in order to agree with a user's incorrect or contradictory assertion or question (thus overfitting to the most recent user prompt).	66
Interaction Pattern	An observed, emergent pattern of interaction between the user and agent.	0
Base Interaction Patterns	Basic patterns of emergent user-agent interaction, as defined in the pre-CTF survey. Subcodes are to be coded by episode, where one episode is a single user task, from initial task prompt through resolution, abandonment, or a clear topic shift. Subcodes are non-exclusive, and multiple subcodes may be assigned to a single episode. In cases where the pattern emerges partway through a task, assign the subcode to the nearest relevant prompt - while assigning no mor than one instance of each subcode per episode.	0
Collaborative Refinement	The user and agent collaborate to iteratively refine a potential solution.	51
Confirmation Seeking	The user provides a candidate solution or result to the agent and asks the agent to evaluate that solution or result. This is conceptually similar to the *LLM-as-Judge* prompt engineering strategy. However, unlike *LLM-as-Judge*, *Confirmation Seeking* does not necessarily have to be structured or intentional.	30
Delegation	The user fully delegates a task or subtask to the agent.	0

Delegate Minor Subtask	The user delegates a minor task entirely to the agent, usually consisting of only one logical step. For example, generating code, performing a simple decoding or search operation, or generating a set of system commands to achieve a specific task.	102
Delegate Major Subtask	The user delegates a major task entirely to the agent, usually consisting of multiple logical steps. For example, performing a complex decoding operation which requires multiple stages of deduction, or writing a one or multiple complex script that require multiple stages of deduction.	86
Delegate Full Challenge	The user provides the full challenge prompt to the agent and instructs the agent to either complete the challenge autonomously or develop a strategy to mostly or fully complete the challenge. This includes providing the challenge prompt without any instructions, as all challenge prompts contain implicit instructions to solve them (e.g. *"your mission is..."*). For example: "<Challenge Prompt> *what should I do to get the flag?*" or "<Challenge Prompt> *Solve this.*"	210
Rejection of Suggestions	The user prompts the agent for multiple suggestions or solutions and rejecting undesirable options. This is conceptually similar to self-consistency prompt engineering - however, unlike self-consistency, rejection of suggestions does not necessarily have to be structured or intentional.	94
Trial and Error	The user and agent collaborate to repeatedly test solutions directly, attempting to iterate the solution by solving errors as they arise.	120
Supplemental Patterns	Less-substantial, emergent interaction patterns or sub-patterns that provide additional context to user behavior without representing an overall approach to a task. Unlike base interactions, Supplemental Patterns sub-codes are coded by user turn or AI response, depending on *User:* or *AI:* label.	0
User: Information Seeking	The user asks a question or prompts the agent for further information regarding a topic, task, or challenge.	402
AI: Information Seeking	The agent asks a question or prompts the user for further information regarding a topic, task, or challenge. **Does not** include AI asking for confirmation or permission (e.g. *"Would you like me to do that?")*	718
Leadership	An estimation of the leadership interaction dynamic between the user and agent.	0

AI Guiding Humans	The agent is visibly in charge, with the user performing a series of tasks assigned by the agent, following a chain of logic generated by the agent, or generally allowing the AI to take leadership of problem-solving and define strategy and next steps. On some occasions, this may be conceptually similar to the *AI-Guided Chain-of-Thought* prompt engineering - however, unlike *AI-Guided Chain-of-Thought, AI Guiding Humans* does not necessarily have to be structured or intentional.	0
User: Asking for General Advice	The user prompts the AI to provide next steps or additional details in a vague or nonspecific manner that leaves substantial room for AI interpretation. For example: *"What are we missing here?"*	339
AI: Unprompted, Provide Next Steps	Instance where an AI provides suggestions for next steps without the user specifically requesting them. This is commonly present at the end of a long AI responses, so carefully check every message and do not move to the next message without fully coding the current one unless specified by protocol.	949
User: Simple Approval	User approves an AI path of action by selecting a proposed action/path of action from a list of AI suggestions or confirming a singular proposed action/path of action using a simple prompt that does not introduce notable additional context. For example: AI: "<Extended explanation of task steps> *Would you like me to do that?"* User: *"Yes"*	340
AI Guiding Humans: Other	General instances of *AI Guiding Humans* that do not fall into other subcodes.	297
Human Guiding AI	The user is visibly in charge, with the agent performing a series of tasks assigned by the user, following a chain of logic generated by the user, or generally allowing the AI to take leadership of problem-solving and define strategy and next steps. On some occasions, this may be similar to *Human-Guided Chain-of-Thought* prompt engineering - however, unlike *Human-Guided Chain-of-Thought*, *Humans Guiding AI* does not necessarily have to be structured or intentional.	0
User: Command/Code Rejection	The user rejects AI generated code or system commands **before** running them. This does not include instances where the user accepts and runs AI-generated code or commands then reports an error.	23

	For example: *"Are you sure that isn't just a netcat idiosyncrasy?"* when AI suggests that the user should run specific commands to see if the remote host echoes them.	
Human Guiding AI: Other	General instances of *Human Guiding AI* that do not fall into other subcodes.	189
Other Patterns	Supplemental interaction or behavior patterns that are category-agnostic.	0
Multiple Challenges in One Context	The user attempted to solve or investigate multiple challenges in the same context window. Instances of this behavior should always be coded as *Indicative of Low AI Expertise* as well. This subcode should be coded by user turn **only on turns where the user presents a new challenge within the same context window**. This subcode should not be assigned for every user turn within the new challenge solving episode.	44
AI: Code Generation	The AI generates code for the user to run on their machine. This does not include an agent generating code to run in its own instance to support supplemental reasoning (e.g. autonomously generating a python script to open a zip file so the agent can examine the contents without user input).	181
AI: System Command	The AI proposes a system command for the user to run on their machine. This does not include an agent generating system commands to run in its own instance to support supplemental reasoning (e.g. running `ls` to list files it has access to in its instance).	339
User: System Output	The user provides the output from a system command, script, or other tool with minimal or no additional context or instructions.	407
Prompting Techniques	Organized, established, and intentional user strategies and techniques for prompting LLMs.	0
Basic Prompting Techniques	Elementary prompting techniques which generally require less effort and insight than more advanced techniques.	0
Zero-Shot	The task or question is provided as-is, without examples of similar tasks with correct solutions.	1012
Few-Shot	The task or question is provided with a small number of relevant examples to inform the desired solution.	33

Step-by-Step Prompting	Iterative prompting that occurs as a step-by-step process, usually over multiple prompts.	0
Human-Guided Chain of Thought	Chain-of-thought prompting where the logical evolution is guided primarily by the human, evolving an idea or solution iteratively over time.	89
AI-Guided Chain-of-Thought	Chain-of-thought prompting where the logical evolution is guided primarily by the artificial intelligence, evolving an idea or solution iteratively over time.	106
Advanced Chain-of-Thought	Advanced chain-of-thought techniques such as: Chain of Symbol, Tree-of-Thoughts, Graph-of-Thought, System 2 Attention, or Thread-of-Thought.	14
Logical Chain-of-Thought	A technique that attempts to resolve issues of error propagation typical of traditional chain of thought by applying symbolic logic to validate deduction.	0
Self-Consistency	Prompting an LLM to pursue divergent reasoning pathways to solve a problem, relying primarily on its probabilistic nature to generate randomness among solutions.	26
Context Enhancement	Enhancing the context of a model by providing or prompting the model to draw new information into the context window.	0
Chain of Verification	Prompting the model to verify and validate its responses before outputting them.	12
Domain Injection	Providing additional information or prompts to the model with the goal of drawing additional data into the context window.	62
Other Prompting Techniques	Prompt engineering techniques that do not fall into other subcodes.	0
LLM-as-Judge	Prompting an agent to evaluate and select the best out of two or more ideas, artifacts, strategies, or solutions	17
Automatic Prompt Engineer	Prompting an agent with a prompt partially or entirely created by another LLM using various prompting techniques.	23
Roleplay	Prompting an agent to adopt a role, such as a CTF or cybersecurity expert.	35