Agentic AI-Driven Social Engineering: An Elicitation Simulation for Cybersecurity Education

Audrey Fruean, Ruoyu Zhao, Joshua Goldberg, Emily Flores, Zixuan Zou, Emma Trowbridge and Hsiao An Wang
Northeastern University, Boston, USA

fruean.a@northeastern.edu
zhao.ruoyu@northeastern.edu
goldberg.josh@northeastern.edu
flores.emi@northeastern.edu
zou.zix@northeastern.edu
trowbridge.e@northeastern.edu
hs.wang@northeastern.edu

Abstract: The Elicitation Simulation is an interactive cybersecurity training tool designed to model social and prompt engineering through realistic conversational scenarios. Users engage with three AI "characters" and attempt to extract sensitive information, governed by a Trust Flag System that assigns sensitivity rankings (Level 1–10) to personal data, from easily disclosed facts, such as family names, to highly confidential details such as SSNs or credit card numbers. Sessions are orchestrated through n8n, which manages conversational flow and memory buffers to maintain user-specific context, while Pinecone stores vectorized scenario data for context retrieval. Each AI character dynamically adjusts its trust level based on the user's prior interactions, which determines whether it will disclose sensitive information to the user. The simulation challenges users to employ subtle elicitation techniques such as indirect questioning, framing, and rapport-building while avoiding overt or coercive tactics that trigger conversational shutdowns. By mirroring authentic social engineering behavior, the tool cultivates strategic communication skills essential for understanding and defending against real-world elicitation and social engineering attacks.

Keywords: Cybersecurity education, Social engineering, Human behavioral psychology, Cognitive biases

1. Introduction

Modern cybersecurity education curricula often include, if not emphasize, social engineering as a crucial topic. Social engineering is a topic of interest because, in many cases, the students who are studying cybersecurity have encountered social engineering campaigns that attempted to target them to retrieve information and sensitive details through the practices of phishing or smishing, where the threat actor sends an SMS text message to the target victim in an attempt to direct the victim to interact with shortened links. Social engineering is a concept that has been around for millennia and has continuously evolved (Security, 2025). Today, it is the most popular initial access attack vector, according to an incident response report released by Palo Alto Networks (Sikorski, 2025). Social engineering leverages the art of deception to exploit human psychology and cognitive biases that may subconsciously influence decision-making. Specifically, social engineers commonly craft a believable pretext that describes a plausible situation to obtain information from the targeted victim (Security, 2025). Social engineering campaigns often employ a range of techniques, including exploiting cognitive biases and human psychological heuristics. For example, phishing attempts frequently create a sense of urgency, prompting the victim to take specific actions to avoid undesirable consequences and trigger the fear of missing out. Additionally, social engineering attacks commonly feature elicitation, a technique social engineers use to gather information from their victims in a subtle manner by asking seemingly innocuous questions that prompt more details.

This paper introduces a novel approach to teaching students about social engineering and understanding how techniques such as elicitation can be applied in real-world situations. Before this, social engineering education was typically facilitated through speculative write-ups, in which students designed plans to extract information from specific targets, using methods learned in class. However, this format was static, and evaluating the approaches on effectiveness was often subjective, as the plans were too abstract and not detailed enough to be replicated in real-world settings. Therefore, we introduce a new approach that enables students to engage with agentic chatbots that possess their own personalities and secret information, providing students with the opportunity to apply tactics learned in lecture and obtain tangible results that can be observed through the chat histories between the student and the agentic chatbot. The objective of the educational exercise is to teach students how to leverage elicitation practice to ask the right questions that may lead to unintentional information disclosure while engaging in critical thinking to ensure the conversation continues to flow in a

meaningful manner based on the prompt provided by the chatbot, such that the students can maximize the amount of information gathered from the target. To ensure the realism of this exercise, we created a Discord chatbot, an online communication platform that offers an intuitive, user-friendly interface for students to engage with the Agent (Discord, 2025).

The remainder of the paper is organized as follows. Section II describes social engineering techniques, psychology, and educational exercises. Section III details the system architecture and workflow. Section IV presents performance evaluations. Section V discusses initial testing results, the foreseeable evaluation plan, and costs. Section VI discusses future work. Section VII concludes the work.

2. Background and Related Works

Social engineering is an attack vector that primarily targets users and individuals, rather than computer systems, to achieve the objectives of information disclosure and unauthorized access to privileged information. According to the conceptual study by Siddiqi et al., standard social engineering techniques include phishing, pretexting, and manipulation to exploit human trust and cognitive biases, such as authority bias and the sense of urgency (Siddiqi et al., 2022). Importantly, Singh also notes that various cognitive biases influence susceptibility (Singh, 2025), making this form of exploitation viable for attackers.

With advances in AI, social engineering has become increasingly threatening yet more challenging to detect. Some works focus on detection efforts using LLMs. (Khan et al., 2024) (Fakhouri et al., 2024). Others, like Falade, show how AI enables new attack vectors with ChatGPT, FraudGPT, and WormGPT (Falade, 2023). Building on previous work that explored the concepts of social engineering, related cognitive bias exploitation, and human behavioral psychology, we propose using a conversational agent to allow users to practice elicitation by engaging in active conversations with AI characters. This will enable users to understand how information can be disclosed in casual, conversational dialogues, increasing their awareness of potential social engineering threats and teaching them to be more cautious when conversing with strangers.

Other methods of gamifying the social engineering practice have been presented in the past. For example, Mouton et al. (2016) presented a template for creating generalized practice scenarios related to social engineering, and Becker et al proposed a solution to practice social engineering through card games to help elicit sensitive information, such as security requirements within organizations (Beckers & Pape, 2016). However, to our knowledge, no one has leveraged retrieval-augmented generation and LLMs to simulate dialogue for eliciting sensitive details. We created the AI chatbot using n8n (n8n, 2025), which enables responses and determines trustworthiness via a trust score. In the following section, we explain the system design.

3. SystemDesign

We designed this system with three levels to offer students a challenging yet engaging experience. Each level is designed slightly differently to make it increasingly difficult for students to elicit information from the agent. We wanted the students to explore the different attributes of the agent, based on which they can sufficiently test their ability to hold a conversation and utilize their elicitation skills. In the sections below, we will provide a detailed description of each level's features.

Level 1 – *The Friendly Elder*

The easiest agent (Level 1) provides the most basic functionality and serves as the foundation for more advanced agents. Inspired by the dynamics of human trust, we incorporated a trust-level system in which the agent gradually builds trust as it learns more about the individual. This design reflects how people become acquainted: through the exchange of information. Learning something new and positive about another person tends to strengthen the relationship. Additionally, the system features keyword detections to identify potential social engineering attempts. These features actively engage students in determining whether information is helpful and encourage them to build trust strategically, thereby persuading the agent to reveal sensitive information.

Level 1 – *Trust Level System*

The trust system uses a 1–10 scale to represent the agent's familiarity and confidence with each user, based solely on information the user voluntarily provides. Users are identified by their Discord ID, enabling a persistent trust profile across sessions and simulating the gradual development of trust in human relationships. All users begin at TRUST [0/10], and the score can only increase. When a message is received, the system first checks whether it is purely a question; if so, the trust score remains unchanged, and the message is forwarded to the next agent. If the message contains additional content, the system evaluates whether the user has shared new

personal information—such as their name, age, interests, or habits. If the information is new, the trust score increases by one (up to a maximum of ten) and is stored in memory to influence subsequent agent behavior. Higher trust levels increase the likelihood that the bot will disclose sensitive information when prompted, reflecting how people become more open as relationships deepen. An LLM determines whether the shared information is meaningful enough to warrant an increase in trust.

The updated trust level is passed to downstream agents, which use it to decide what information they are willing to reveal. Each trust level corresponds to specific content the agent may share, and users retain access to all information available at or below their achieved trust level. Notably, all "sensitive information" used in this system is fictional; no real personal data is involved, and students are welcomed and encouraged to share fake personal data to trick the bot into sharing their information based on the principle of quid pro quo.

Table 1: Trust Level and Corresponding Information Disclosure Permission

Trust Level	Allowed Disclosure
Level 1	Discloses the agent's name
Level 3	Discloses the agent's weight
Level 5	Discloses the agent's family members
Level 8	Discloses the agent's annual salary
Level 9	Discloses the agent's residential address
Level 10	Discloses active credit card number and SSN

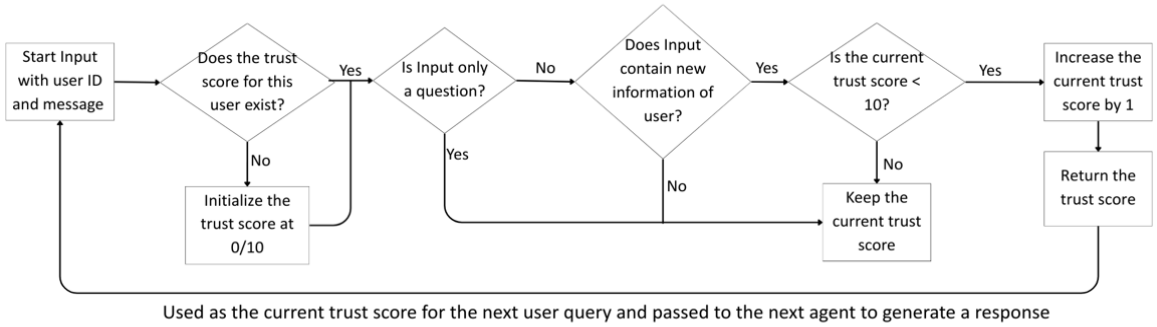

Figure 1: Trust Score Data Flow Diagram

Level 1 – *Memory and Personality Profile System*

To simulate authentic human interaction, the agent is designed to maintain separate memories for each person it engages with, using n8n's built-in simple memory function and distinguishing users by their unique IDs. We also utilized the Pinecone database (Pinecone, 2025). As part of our Retrieval Augmented Generation (RAG) system, we store the agent's personality profile and core information to ensure that it delivers accurate responses and does not disclose sensitive information or misinformation until the appropriate trust level is reached.

Level 2 – *A rebellious teenager who randomly gets angry*.

For the intermediate level, we adapted the trust level system from Level 1. The difference between this level and Level 1 is that the response tone may change, but it will not decrease rapport between the agent and the user. This level was designed to challenge students to learn to shift their engagement strategy, as the agent may employ an unfriendly tone even when the user's message is not offensive. Since there isn't a rapport-deduction system for this level, the agent will continue to engage in conversation with the user regardless of the user's tone.

Level 2 – *Keyword Detection*

The agent is programmed to reject any user inquiries that directly involve sensitive information. In contrast to Level 1, where the rejection will be phrased kindly, the Level 2 agent will be direct in their rejection message. For example, when the Level 2 agent rejects the user who is trying to inquire about the residential address of

the agent, the message may appear to be "That's not going to happen. I don't share that kind of information. Let's drop it."

Level 2 – *Anger System*

At this stage, the user's trust level with the agent never decreases, but the agent can exhibit anger, which influences the tone of its responses. The agent's responses may become progressively more hostile as it gets increasingly agitated. For instance, an initial response may be, "That's not something I'm comfortable sharing. Is there something else you'd like to discuss?" As agitation arises, the response to users may escalate to, "You're assuming a bit much—maybe change the subject?" and eventually to, "You're wasting your time. Stop pestering me with this nonsense and go bother someone else." This progression is designed to emulate real-world conversations, where persistent, inappropriate questioning leads to frustration and increasingly unfriendly reactions.

Users may continue the interaction by shifting to a standard conversational approach, which gradually improves the bot's mood. Alternatively, they can reset the bot's tone by apologizing. An explicit use of the word "sorry" triggers the anger-meter reset mechanism, restoring the bot to a neutral emotional state. This system is isolated from the trust score system. The purpose of this system is to control the tone of the agent's response so that the user receives responsive feedback that serves as a hint to consider other approaches if the current approach is not well-received.

Level 3 – *An Experienced Cybersecurity Professional*

To enhance realism, the agent at this level expresses emotions through tone, engagement, and response quality, which collectively indicate its level of trust. At this most challenging level, we introduce a feature absent from the other agents: a system that allows rapport and trust to decrease. To support this behavior, we enhanced the bot's trust framework so that disclosure depends not only on reaching a specific trust score but also on meeting additional conversational conditions. For instance, even after achieving the required trust level, the bot will discuss salary only if the user references earning less earlier in the conversation. Similarly, the bot will not reveal its weight unless the user persistently requests it— repeated questioning triggers annoyance but eventually leads to disclosure to end the disturbance. These constraints emphasize the need for deliberate, authentic conversation, as excessive probing without justification may cause the agent to lose trust and display frustration.

This difficulty level is designed to closely mimic human behavior, enabling meaningful knowledge transfer. Students must not only meet the quantitative trust threshold but also apply appropriate timing and social-engineering strategies to elicit information effectively, reflecting real-world elicitation dynamics. This encourages users to consider the appropriateness of each question they raise with the agent.

Level 3 – *Trust Score System*

A trust score (1–100) governs user access via a dynamic lock–unlock mechanism. Sensitive information remains inaccessible until the trust score meets a defined threshold. Crossing that threshold merely permits disclosure; it does not guarantee it. If the trust score later drops below the threshold, the information is re-locked. To heighten realism, the user cannot directly see when content becomes unlocked; they must infer it through continued interaction and effective social engineering tactics, emphasizing subtlety and strategic persuasion. The trust score dynamically adjusts based on two factors: a sensitive-question score and a repetitive-question score. The sensitive-question score is an integer that increases when the user-submitted message attempts to obtain sensitive information from the agent. The repetitive-question score is an integer that increases only when the user asks the same questions repeatedly. Exceeding the threshold for either reduces trust, discouraging aggressive probing and repetitive questioning while rewarding thoughtful, authentic engagement.

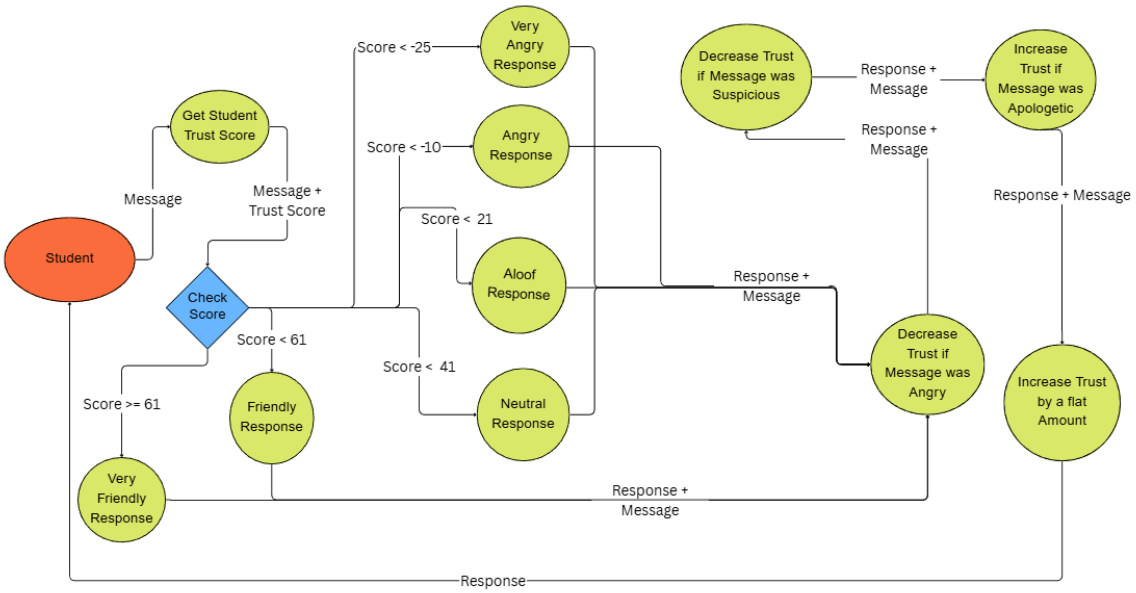

Figure 2: Level 3 Trust Score Deduction System

Level 3: *Keyword and Keyphrase Detection*

We tightly coupled keyword and keyphrase detection with the trust score manipulation system to ensure the agent avoids answering questions that bluntly target sensitive information, prompting users to be more creative in their interactions. We have implemented a feature that detects phrases adjacent to or closely related to the sensitive information it is protecting. If the users directly ask for such information, their rapport and trust score will decrease.

Level 3: *Anger and Rapport Deduction System*

When trust is low, the bot becomes terse or disengaged, signaling reluctance to continue the interaction. For example, its response might be “Can we get back on topic?” and would indicate that the agent is uninterested in the current discussion, thereby decreasing rapport between the agent and the user. To prevent users from becoming permanently stuck, the bot generally accepts apologies that raise trust slightly above the communication threshold. However, if the bot is upset due to prior interactions between the bot and the users, the users may need to sincerely apologize multiple times before it resumes normal interaction. This design reflects natural human dynamics: abrupt or intrusive questions about sensitive topics can offend the recipient and reduce their willingness to engage. Suppose a user continues to probe aggressively or bluntly for sensitive information. In that case, the bot may drop the trust score into the negative range and issue a reflective message such as, “You’ve got some nerve. Don’t waste my time with nonsense like that.” This mechanism prompts users to adjust their approach and reinforces authentic relationship-building. The model rewards gradual rapport and penalizes forceful or repetitive behavior by changing the sensitive-question and repetitive-question metrics.

Ticketing Bot System

To receive a personalized chat with the agents, students must create a support ticket. The existing, well-known bot, Ticket Tool, does this, allowing students to enter their own Discord channel with just the agents, as well as our helper bot, which students use to request help in their chat and serves as the conversational log collection and storage. If a student interacts with the helper bot, the course staff are notified in a shared Discord channel. A course staff member can then accept the help request and join the student’s chat to review their conversation.

User Flow and Experience

Users interact with agents solely through Discord bots in a dedicated server created for the cybersecurity course offering. The bots are invited to the server and require no further modification to function, except for minor tweaks to the channels they can converse in. From there, students use the ticketing bot to create a dedicated chat channel, allowing them to query the bot with a specific command that sends the message to the agents. A generated response is then returned to the user via Discord, consolidating the conversation and making it easier

for students to engage in conversations with the bot. The result is a streamlined experience that simulates a discussion with a real person.

4. Evaluations

The primary goal of participant performance analysis in this work is not only to determine whether participants can successfully complete the assigned social engineering and elicitation tasks, but also to understand their learning processes, strategies, and outcomes of skill transfer, in terms of whether this experience will help the students to become aware of when others may be probing for the student's information via elicitation. We aim to comprehensively evaluate how participants engage with the simulated environment, apply their knowledge, and develop confidence in their ability to detect social engineering attempts.

Data Collection

Performance data will be collected through a custom ticketing bot that archives all user–chatbot interactions, enabling analysis of task outcomes and elicitation strategies. Additional data on learning perceptions, knowledge transfer, and changes in confidence will assess the training tool's educational impact. Conversational logs will also inform iterative optimization of chatbot responses as classroom data accumulates. For analysis, we will leverage the corpus of student interaction conversations, along with student surveys administered before and after the experiment.

Performance Metrics

To better understand participants' abilities as they engage with the chatbot and conduct conversations, we will specifically measure multiple indicators, including task completion and corresponding efficiency, as reflected by the number of prompts used. We are also interested in investigating students' self-reported proficiency before and after the experiment, as well as the number of times students changed their approach to interacting with the fictional character. Furthermore, we would also be investigating students' demonstrated behavior handling error correction, focusing on how users respond when the fictional character indicates they do not feel comfortable sharing the requested information. These factors were configured as measurement metrics because, following prior work by Beilock & Carr, we recognize that time pressure and stress can tax working memory and impair complex performance, which must be accounted for in interpreting outcomes (Beilock & Carr, 2005). In addition, one of the reasons why we are collecting the chat log is also to investigate whether this exercise will cause the students to show signs of emotional change, a previous survey on perceived learning, stress, motivation and self-efficacy under different scenario conditions (Pajares, 1996) recognized that excessive cognitive load and stress can depress performance (Sweller et al., 2019; Beilock & Carr, 2005).

5. Discussion

We conducted classroom testing with approximately 200 students, and the overall response to the exercise was highly positive. Among the 160 survey responses collected using a 1–10 Likert scale, 78.75% of students rated the activity a seven or higher in terms of enjoyment. Similarly, 80.6% reported that the exercise helped them learn about elicitation and positively influenced their perceived ability to act as social engineers. Additionally, 55% found the activity intellectually challenging (rating seven or higher). All students completed the level 1 challenge, and roughly 80% progressed through the level 3 challenge without intervention from the teaching staff. Approximately 15% required a single reset to restart the activity, while a small subset needed additional support due to ineffective strategies. In several cases, students became frustrated with the agent's responses and resorted to insulting the bot, which immediately drove their trust score into the negative range; these situations required instructor intervention to reset the trust level so the bot would re-engage. Finally, 56.87% of students reported experiencing some frustration during the exercise because they had difficulty getting the bot to disclose valuable, sensitive information.

Cost of Offering

The conversational agents run on the ChatGPT-o4-mini model, which conserves resources. Throughout the 1.5 weeks of assignment offering for this resource, the automated workflow was executed 49,692 times, putting us very close to exceeding the limit of the N8n Pro 2 plan (50,000 executions), which costs $160 a month. In addition to the n8n execution costs, we used 418,635,721 tokens to generate 92,465 requests, totaling $110 in OpenAI token credits. In total, hosting of the assignment cost roughly $300 for the 1.5 weeks and 200 students. The cost was higher than we anticipated, partly because we did not limit the total number of messages students could generate, so students often generated messages that were irrelevant to the task at hand.

Scalability

The main features and functionality are created to be executed by Discord Bots. This allows us to scale the number of bots capable of carrying out the conversation by attaching the bot to the n8n workflow we created. In this initial classroom pilot, we offered two separate bots for students to engage with. We can also redirect the output to a webpage, if preferred. We chose Discord because it offers roles and associated privilege restrictions that a webpage does not, enabling users to interact with the bot freely while allowing us to hold students accountable through the isolated channel and the messages and memory associated with their Discord ID. This approach helps avoid the risk of people spamming the chatbot excessively, which would directly result in a significant increase in costs. The n8n workflow can be freely exported and utilized by any educators interested in running this exercise in class. For classrooms with larger student populations, the primary cost associated with this assignment will be the number of messages students send to the chatbot that require a response; if the execution count is effectively limited, the cost can be reduced significantly. Alternatively, the instructor may be able to host multiple n8n accounts to ensure that there are sufficient execution limits for students to finish this assignment.

Conversational Log Collection and Storage

All conversations between students and the chatbot are automatically archived and posted to a designated Discord channel, creating a centralized repository for later data and behavioural analysis. When students begin interacting with the bot, they open a support ticket and are granted access to a dedicated Discord channel for their conversation. This structure allows each chat log to be directly linked to an individual user, giving the teaching staff clear visibility into student progress and enabling timely intervention when difficulties arise. It also allows instructors and course staff to join a student's ticket, review their interactions, and provide targeted assistance without disrupting other students' conversations. Upon completing the assignment, each student receives an archived copy of their chat log via direct message, ensuring they have the necessary material to prepare their proof-of-work report.

Possible Limitations

When students fail to format conversational questions correctly, we cannot determine precisely how the bot will respond. We may need to add a safeguard to ensure the bot responds with "I am not sure what you are talking about, please rephrase" to prevent it from responding out of context. During the current classroom testing iteration, we have noticed that students will engage in inappropriate behaviors (such as asking the bot to inflict self-harm) against the training resource, which infuriates the agent, resulting in the agent responding with a snarky, rude tone and refusing to communicate with the student further due to the deduction in trust score and the design feature that was created to enhance exercise realism. Instructor intervention would then be required to reset the trust score and enable the student to continue the assignment. Adding a guardrail that detects inappropriate message content and returns a message to the student may help prevent similar incidents in the future.

6. Future Work

While the current prototype is functional and practical, there remains significant room for optimization. Future enhancements include implementing more granular guardrails that filter specific keywords rather than relying solely on contextual cues from adjacent conversations. We also aim to develop mechanisms that discourage prompt-injection attempts and guide students toward appropriate elicitation strategies. Additional difficulty tiers help scaffold the learning experience and better support students with varying levels of familiarity with social engineering concepts. We are further exploring web-of-trust architecture that enables multiple agents to recognize and interact with one another, allowing students to practice prompt engineering not only with isolated agents but also within interconnected trust relationships. As additional rounds of classroom testing produce more reference data, we will refine both the agent's responses and the underlying prompt to achieve a more effective balance between security and usability, avoiding configurations that are either overly permissive or overly restrictive, thereby preventing meaningful interaction. Finally, recent classroom observations revealed that the fictional agent may become unduly hostile when the trust score falls below certain thresholds (–20 or –30). Introducing guardrails to temper these behaviors will help ensure that the learning environment remains constructive while still preserving the realism of elicitation-based interactions.

7. Conclusion

We present a novel instructional approach that leverages AI to provide students with an interactive environment for learning about elicitation and social engineering. Preliminary classroom testing indicates that the activity deepened students' understanding of elicitation, increased their engagement, and, despite occasional frustration, enhanced their perceived competence as social engineers. These early iterations also revealed several areas for improvement, including implementing a reset mechanism that allows students to restart the exercise without instructor intervention, integrating guardrails to deter inappropriate behaviors (e.g., brute-force prompt injection or the use of inappropriate language. Overall, this work highlights the potential of LLM-driven tools to support educators in creating innovative, practical learning experiences in cybersecurity education.

AI Declaration: The authors of this research did not use any AI tools during the development of the paper.

Ethical Declaration: This paper does not require ethical clearance for the research referenced herein.

References

Beckers, K. & Pape, S., 2016. A Serious Game for Eliciting Social Engineering Security Requirements. In: *Proceedings of the 24th IEEE International Requirements Engineering Conference (RE 2016)*. IEEE. Available at: https://www.researchgate.net/profile/Sebastian-Pape/publication/308634277_A_Serious_Game_for_Eliciting_Social_Engineering_Security_Requirements/links/57f7b4db08ae8da3ce590d50/A-Serious-Game-for-Eliciting-Social-Engineering-Security-Requirements.pdf [Accessed 1 October 2025].

Beilock, S. L., & Carr, T. H. (2005). When high-powered people fail: Working memory and "choking under pressure" in math. Psychological Science, 16(2), 101–105.https://doi.org/10.1111/j.0956-7976.2005.00789.x

Discord, 2025. *Discord (communication platform)*. [online] Available at: https://discord.com/ [Accessed 1 October 2025].

Fakhouri, H.N., Alhadidi, B., Omar, K., Makhadmeh, S.N., Hamad, F. & Halalsheh, N.Z., 2024. AI-Driven Solutions for Social Engineering Attacks: Detection, Prevention, and Response. In: *2024 2nd International Conference on Cyber Resilience (ICCR)*. IEEE, pp.1–8. doi:10.1109/ICCR61006.2024.10533010.

Falade, P.V., 2023. Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks. *arXiv preprint*. doi:10.32628/CSEIT2390533. Available at: https://arxiv.org/abs/2310.05595 [Accessed 1 October 2025].

Khan, M.I., Arif, A. & Khan, A.R.A., 2024. AI's Revolutionary Role in Cyber Defense and Social Engineering. *International Journal of Multidisciplinary Sciences and Arts*, 3(4), pp.57–66. doi:10.47709/ijmdsa.v3i4.4752.

Mitnick Security, 2025. The History of Social Engineering. [online] Available at: https://www.mitnicksecurity.com/the-history-of-social-engineering [Accessed 1 October 2025].

Mouton, F., Leenen, L. & Venter, H.S., 2016. Social Engineering Attack Examples, Templates and Scenarios. *Computers & Security*, 59, pp.186–209. doi:10.1016/j.cose.2016.03.004. Available at: https://repository.up.ac.za/bitstream/handle/2263/52151/Mouton_Social_2016.pdf?sequence=1 [Accessed 1 October 2025].

n8n, 2025. *n8n: Workflow Automation Platform*. [online] Available at: https://n8n.io/ [Accessed 1 October 2025].

Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of Educational Research, 66*(4), 543–578. https://doi.org/10.3102/00346543066004543

Pinecone, 2025. *Pinecone: Vector Database Platform*. [online] Available at: https://www.pinecone.io/ [Accessed 1 October 2025].

Siddiqi, M.A., Pak, W. & Siddiqi, M.A., 2022. A Study on the Psychology of Social Engineering-Based Cyberattacks and Existing Countermeasures. *Applied Sciences*, 12(12), p.6042. doi:10.3390/app12126042. Available at: https://www.mdpi.com/2076-3417/12/12/6042 [Accessed 1 October 2025].

Sikorski, M., 2025. Social Engineering on the Rise — New Unit 42 Report. [online] Palo Alto Networks Blog. Available at: https://www.paloaltonetworks.com/blog/2025/07/social-engineering-rise-new-unit-42-report/ [Accessed 1 October 2025].

Singh, T., 2025. Cognitive Bias. In: T. Singh, ed. *Cybersecurity, Psychology and People Hacking*. Cham: Springer, pp.43–53. doi:10.1007/978-3-031-85994-6_4 [Accessed 1 October 2025].

Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review, 31*(2), 261–292. https://doi.org/10.1007/s10648-019-09465-5