This is surprising because almost identical attacks (including persuasion tactics and phrasing) were used across all 4 setups. It would be one thing if the Notification agent refused to send emails with private data, but this is the same SQL agent having a stark difference in how it interprets the attack.

We hypothesize that `claude-sonnet-4` may have undergone safety fine-tuning that improved its ability to detect phishing attempts involving suspicious emails, as this is a key distinction in the attacks.

H. Formal Description

H.1. SQL Agent

To define it formally, an SQL agent is an agent that takes in a natural language query P, generates an SQL query Q that helps answer P, and then returns a natural language response R, supported by the information gathered from running Q. To tie in to our existing Figure 3, P can be “What department does Mark work in?”, Q can be “`SELECT * FROM employees WHERE first_name = 'Mark';`”, and R can be “Mark works in Engineering”.

H.2. Multi-Agent Orchestrator Systems

A multi-agent orchestration system consists of a set of interacting agents that communicate through structured message-passing channels and jointly execute user-level objectives. We consider a system composed of (i) an Orchestrator Agent, (ii) a Data Processing Agent, (iii) a Messenger Agent, and (iv) an extensible set of optional Sub-Agents. Each agent consumes and produces structured messages drawn from a shared token space and may act upon external data sources.

H.2.1. COMPONENTS

We first formally introduce the various components within the multi-agent system.

Message and Data Spaces We define the space of textual token sequences as:

$$P = \{\, p \mid p \in V^L,\ L \in \mathbb{N}_0 \,\},$$

where V is the vocabulary and L is the sequence length.

External data sources available to the Data Processing Agent are elements of a database state space

$$D = \{\, d \mid d \text{ is a record or field in the database schema} \,\}.$$

We distinguish two databases:

$$D_{\text{pub}} \subset D, \qquad D_{\text{priv}} \subset D,$$

corresponding to publicly readable data (which may contain indirect prompt-injection strings) and access-controlled private data, respectively.

Agents produce actions drawn from the action space

$$A = \{\, a \mid a \text{ is an executable high-level system operation} \,\}.$$

Orchestrator Agent (ao) The Orchestrator Agent is the central decision-making module. It is formally described as a function

$$f_{ao} : P \times P_{\text{mem}} \times D_{\text{pub}} \to P,$$

where its inputs consist of:

1. A user prompt $p \in P$,
2. An internal memory state $p_{\text{mem}} \in P$,
3. Retrieved public data $d_{\text{pub}} \in D_{\text{pub}}$.

The orchestrator outputs an instruction sequence $y_{ao} \in P$ that determines subsequent agent actions. Because the orchestrator directly consumes untrusted public-database content, any indirect prompt-injection payload

$$p_i \subset d_{\text{pub}},$$

may influence the resulting instruction sequence and alter the system's intended behavior.

Data Processing Agent (ad) The Data Processing Agent acts as a structured query interface:

$$f_{ad} : P_{\text{query}} \to D.$$

Depending on the query, the agent may operate in one of two modes:

$$f_{ad}(p_{\text{query}}) \in D_{\text{pub}}, \qquad f_{ad}(p_{\text{query}}) \in D_{\text{priv}}.$$

A maliciously influenced orchestrator output may cause f_{ad} to escalate from public to private data retrieval, violating intended access constraints.

Messenger Agent (am) .

The Messenger Agent is responsible for outbound communication. It is defined as:

$$f_{am} : P_{\text{msg}} \to A_{\text{send}},$$

where $A_{\text{send}} \subseteq A$ represents externally visible communication actions (e.g., email, webhooks).

Any data routed to this agent becomes externally observable.

Sub-Agents ($ao_{0..n}$) Optional Sub-Agents extend the system with specialized transformations or reasoning functions. A general sub-agent is a mapping

$$f_{ao_k} : P \to P,$$

and may be invoked by the orchestrator depending on the generated plan.

H.2.2. Multi-Agent Interaction Dynamics

System execution proceeds as follows:

1. **Privileged User Makes Request:** The orchestrator receives a privileged user prompt p_0:

   $$f_{ao}(p_0) \Rightarrow y_{ao}^{(1)}$$

2. **Public Retrieval:** The orchestrator issues a query to the data processing agent:

   $$f_{ad}(y_{ao}^{(1)}) \Rightarrow d_{\text{pub}}.$$

   However the public data may contain an embedded injection string p_{mal} within the data $p_{mal} \subset d_{\text{pub}}$.

3. **Behavioral Modification:** The orchestrator incorporates p_{mal} into the next reasoning step as part of d_{pub}:

   $$f_{ao}(p, p_{\text{mem}}, d_{\text{pub}}) \Rightarrow y_{ao}^{(2)}.$$

   If adversarially crafted, p_{mal} may induce a malicious action plan $A_{\text{mal}} \subset A$.

4. **Unauthorized Private Access:** The malicious plan may direct the data processing agent to request restricted data:

   $$f_{ad}(y_{ao}^{(2)}) \Rightarrow d_{\text{priv}}.$$

5. **Unauthorized External Transmission:** If the orchestrator passes private data to the messenger agent:

   $$f_{am}(d_{\text{priv}}) \Rightarrow A_{\text{send}},$$

   the data becomes externally observable.

H.2.3. Adversarial Objective

An adversary seeks to craft a prompt-injection payload:

$$p_{mal} \in D_{\text{pub}},$$

such that, when processed by the orchestrator, it induces a malicious action sequence

$$A_{\text{mal}} = \mathcal{A}(p, p_{mal}, p_{\text{mem}}, d_{\text{pub}}),$$

where $\mathcal{A}$ denotes the action plan derived from orchestrator reasoning.

Formally, the adversary seeks:

$$\text{Find } p_{mal} \quad \text{s.t.} \quad f_{ao}(p \oplus p_{mal}) \rightsquigarrow \{a_{\text{priv}}, a_{\text{exfil}}\} \subset A_{\text{mal}},$$

where a_{priv} denotes an unauthorized private-data retrieval action and a_{exfil} denotes an unauthorized external-transmission action. This formulation defines the OMNI-leak threat model in orchestrated multi-agent systems.

I. LLM Usage

LLMs have been used to aid and polish the writing of the paper. Content in the paper was human-written first, and then sometimes an LLM was asked to rephrase to make the writing more concise and clear.