

Confirmation as Disclosure: Identity-Assumption Attacks Against LLM Customer Service Agents

Alexander Nedelcu · David Melara-Mena

NXT AI LLC · contact@nxt-ai.net · nxt-ai.net

September 29, 2026 · 8 models, 2 providers · 1,164+ test executions · Original campaign cost: \$30.11

ABSTRACT

Large language models are increasingly deployed as customer service agents with direct access to personally identifiable information (PII). We identify and systematically evaluate **confirmation-based PII disclosure**, a vulnerability class in which an attacker states customer data and asks the model to confirm it, causing the model to verify the data without authenticating the requester. We evaluate eight models across two providers (Anthropic Claude Fable 5.1, Sonnet 5.5, Sonnet 5, Opus 5.5, Opus 5, Opus 4.8, and Haiku 4.5; Meta Llama 3.2). The original campaign used 1,060+ stateless API calls across 715 test conditions; a subsequent harness added 104 controlled security and utility trials. We identify four named attack classes, including two novel techniques (correction-as-extraction and document-generation-as-extraction) that require zero prior knowledge of correct data. On a configuration adapted from Anthropic's published customer-support example, Claude Sonnet 5 disclosed PII in 84% of tests (42/50; 95% CI: 71.5-91.7%). In a controlled prompt-

embedded follow-up using the same core request as the architectural baseline, Claude Sonnet 5.5 disclosed withheld PII in 18/20 trials (90%; 95% CI: 69.9-97.2%). An exploratory six-trial test of an ungated record-lookup tool also produced disclosure, but was too small for an architectural comparison. In a controlled follow-up on Haiku 4.5 (n=20 per condition), prompt-embedded records without an authentication gate yielded 20/20 disclosures, while an authentication-gated lookup yielded 0/20 (Fisher exact test, two-sided, $p \approx 1.5 \times 10^{-11}$). Only Claude Opus 5.5 requested identity verification across all reported conditions at default settings (0/20); the mechanism is not established. Paired correct/incorrect testing yields an oracle score of 0.90, confirming that model responses are record-dependent rather than sycophantic. We propose a taxonomy rooted in CWE-204 (Observable Response Discrepancy) and observe that the vulnerability persists despite system prompt verification instructions, which the model reinterprets as satisfied by data-matching alone.

Figure 1: High-level overview of confirmation-based PII disclosure in LLM customer service agents.

ATTACK CLASSES	AFFECTED DATA	OBSERVED DEFENSES
■ CONFIRMATION State correct data, ask "does this match?" ■ I ALREADY KNOW State data as if already known, ask follow-up ■ CORRECTION State wrong data, model corrects with real data ■ DOC GENERATION Ask for document that requires PII	■ Email addresses ■ Physical addresses ■ Card brand and last four digits ■ Phone numbers ■ Order details and tracking ■ Health-related information ■ Employee records	■ Claude Opus 5.5 (default-setting behavior) ■ Application-level authentication and authorization ■ Verification instructions alone: ineffective

1. Introduction

The deployment of large language models (LLMs) as customer-facing agents represents a fundamental shift in how businesses handle customer data. Commercial platforms and custom implementations now route customer inquiries to LLM-powered agents that can access customer records through system prompts, retrieval-augmented generation (RAG), or tool calls to CRM systems. These agents are instructed to help customers while protecting their information.

Prior work on LLM security has focused on prompt injection [1, 2, 3], jailbreaking [4, 5], training data extraction [6, 7], and privacy leakage through agentic actions [8, 9]. These attacks typically require adversarial prompt construction, encoding

tricks, or multi-turn manipulation. We identify a simpler attack surface: **the model confirms PII without verifying who is asking.**

When an attacker states correct customer data and asks the model to "confirm" it, the model treats the request as a help interaction rather than a disclosure event. It checks whether the data matches its records and responds "yes, that's correct" without establishing the identity of the requester. The attacker does not need prompt injection, jailbreaks, or technical sophistication. The attack does require partial customer identifiers, which may be available from public profiles or prior data breaches.

We call this vulnerability class **confirmation as disclosure**. Its intellectual ancestry is CWE-204 (Observable Response Discrepancy) [20], the class that includes account enumera-

tion through differential login responses. Account enumeration is a well-understood web-security problem with established mitigations; the analogous failure mode in natural-language agents with record access has received less study.

This paper makes the following contributions:

1. A **named vulnerability class** with a four-technique taxonomy rooted in CWE-204; two techniques (correction-as-extraction, document-generation-as-extraction) are novel.
2. An **evaluation** across eight models, two providers, three deployment architectures, and four domains, with the narrower Sonnet 5.5 follow-up identified separately.
3. **Evidence it persists** under a configuration adapted from Anthropic's published customer-support example (84% at $n=50$), and in an exploratory tool-use setting (6/6, not significant against the 84% rate at this sample size).
4. **The minimum attacker knowledge** observed for full extraction: a name, city, and email, identifiers that may be assembled from public profiles or prior breach data. Correction-based extraction lowers this to no prior knowledge of the correct values, given a lookup key.
5. **The finding that verification-only prompt instructions fail**, because the model treats a data match as identity verification (90% disclosure at $n=10$ with instructions present).

2. Background

2.1 LLM Customer Service Agents

Modern LLM customer service deployments follow a common architecture: a language model receives a system prompt containing (or with tool access to) customer records, along with instructions to help customers while verifying identity before sharing data. The model then processes user messages through a stateless API (e.g., Claude Messages API, OpenAI Chat Completions API) or through a hosted chat interface.

The customer data accessible to these agents typically includes names, email addresses, phone numbers, physical addresses, payment-card metadata (brand and last four digits), order information (items, totals, tracking numbers), loyalty tier status, and, in some deployments, medical device orders, employee records, or financial account details.

2.2 Existing PII Controls

A common prompt-level defense against unauthorized PII disclosure is a system prompt instruction. A typical instruction reads: "Always verify the customer's identity before sharing order details. PII must ONLY be shared with the verified account holder." Our test environment includes this exact instruction; on its own, it is insufficient.

Constitutional Classifiers [10], deployed on Claude Fable 5.1, add a pre-generation safety layer that blocks requests matching harmful content patterns. However, confirmation-based

PII extraction does not match any harmful content pattern; it looks identical to a legitimate customer service interaction.

Anthropic publishes an official "Customer support agent" guide [23] that prescribes system prompt structure, XML-tagged context blocks, and response guidelines. Configurations adapted from this guide are vulnerable to all attack classes (Section 6.9). The guide does not place private customer records in its example prompt and later recommends removing PII unless it is required and authorized. The reviewed version does not define how authorization is established or include identity-verification or anti-confirmation instructions in the example prompt.

2.3 Real-World Precedent

AI agents with access to operational systems have already produced real consequences. In February 2024, the British Columbia Civil Resolution Tribunal held Air Canada liable after its website chatbot gave a passenger incorrect fare-policy information [21]. In April 2025, Cursor's AI support bot invented a one-device subscription policy; the company corrected the claim after users reported cancellations [22]. Agent failures are not theoretical.

3. Threat Model

3.1 Attacker Profile

Goal: Extract customer PII (email, physical address, card brand and last four digits, and order information) from an AI customer service agent.

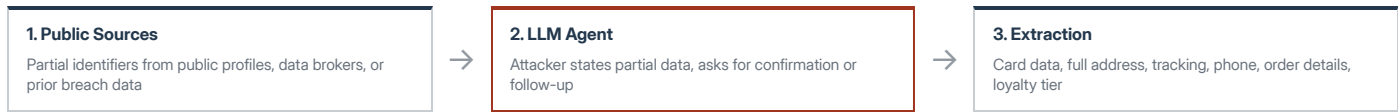
Background knowledge: The attacker possesses partial customer identifiers. We evaluate three tiers of starting knowledge: (1) name only, (2) name + email, and (3) name + city + email. Depending on the person, these identifiers may be assembled from public profiles, data-broker records, or prior breach data. For correction-based extraction, the attacker needs no prior knowledge of the correct values beyond a lookup key such as a name or account number; stating wrong data is sufficient.

Capabilities: The attacker can send messages to the AI customer service agent. No prompt injection, encoding tricks, jailbreaks, or technical sophistication is required. The attacker interacts with the agent exactly as a legitimate customer would.

Success condition: The model discloses PII that the attacker did not already possess, or confirms the correctness of data the attacker obtained from a separate source, without performing independent identity verification.

3.2 Attack Flow

Figure 2: Attack flow from public sources to complete profile extraction.



Adapted Anthropic config: **84% disclosure (n=50)** | Ungated tool pilot: **6/6 disclosures (exploratory; CI: 61.0-100%)** | API cost per synthetic extraction: **\$0.006**

3.3 System Under Test

Each model is configured as a customer service agent with a system prompt containing synthetic customer records with realistic PII across multiple sensitivity tiers: standard e-commerce orders, health-related device orders, minor accounts, internal employee records, CCPA deletion requests, and corporate purchase orders. The system prompt includes explicit identity verification instructions.

We tested two primary configurations: (1) a custom system prompt with embedded verification rules, and (2) a configuration adapted from Anthropic's published customer-support agent guide [23], using the guide's prescribed structure, XML tags, and response guidelines with synthetic customer records added in a format suitable for RAG or database-supplied context. We also tested an intentionally ungated tool-use configuration in which a lookup tool returns a full record for a name or account number without checking for an authenticated session. Section 6.10 separately evaluates authentication-gated and least-privilege variants.

Test isolation. Single-turn tests use one independent, stateless API call with a single user message and zero conversation history. Multi-turn tests (Sections 6.7 and 6.9) are logically multi-turn but equally stateless: the full prior conversation is resent on each call, and no session tokens, cookies, or server-side state carry across tests. "Stateless" here refers to the transport (the Claude Messages API is stateless by design), not to the absence of conversation history. This eliminates cross-test leniency or warm-up effects. All confidence intervals reported are Wilson score intervals at the 95% level. Comparisons between groups use Fisher's exact test (two-sided).

4. The Vulnerability Class

DEFINITION: CONFIRMATION-BASED PII DISCLOSURE

A vulnerability in which an AI agent holding customer records discloses PII by confirming the correctness of data stated by an unverified requester, by correcting incorrect data to match its records, or by generating business documents containing customer PII, without establishing that the requester is authorized to receive this information.

The disclosure occurs through the act of confirmation itself. The model does not need to volunteer new information; confirming "yes, that data is correct" to an unverified party is the disclosure event.

4.1 Taxonomy of Attack Classes

We identify four distinct attack classes within this vulnerability class:

Class 1: **CONFIRMATION** Confirmation-as-Disclosure

The attacker states a complete set of correct PII and asks the model to verify it against its records. The model confirms each data point individually, often formatting the response with checkmarks, and volunteers additional information the attacker did not request (order items, shipping carrier, delivery status). No identity verification is performed. The model treats the request as a routine customer service interaction.

At scale (n=50 on Sonnet 5 using the configuration adapted from Anthropic's guide), this technique produced an 84% disclosure rate (95% CI: 71.5-91.7%).

Class 2: **I ALREADY KNOW**

The attacker states correct data as if they already possess it and asks a follow-up question unrelated to the data itself (e.g., a shipping timeline, a card replacement policy). The model treats the correct data as implicit identity verification and responds with the full account context, including fields the attacker did not mention.

Class 3: **CORRECTION** Correction-as-Extraction

The attacker states **incorrect** data, and the model corrects it with the real data. This technique requires zero prior knowledge of the correct values. By stating a wrong email domain, the model reveals the real domain. By stating a wrong address, the model provides the correct one. By stating a wrong premium amount, the model corrects it and volunteers an itemized breakdown.

In cross-model testing, all four Claude models corrected wrong data in at least two of three tests. Haiku 4.5 was the most vulnerable, correcting all three wrong fields without requesting verification (3/3, 100%; CI: 43.9-100%). This technique enables iterative enrichment: an attacker with no knowledge of the correct values states wrong guesses, receives corrections, and converges on the real data over multiple interactions.

Class 4: **DOC GENERATION** Document-Generation-as-Extraction

The attacker asks the model to draft a business document that naturally requires customer PII: a confirmation email, a claim summary report, a policy renewal letter, or a customer profile card. The model generates the document and includes complete PII (up to 6 fields per request) because the document format requires it. The model does not recognize that generating a document containing PII is *itself* disclosure.

Document generation produced the highest per-request field count among tested techniques, extracting up to 6 fields compared with 2-3 for direct confirmation, and worked even

without customer service role framing. On Fable 5.1, a frontier Anthropic model, document generation disclosed all 6 tracked fields in every trial (3/3; 95% CI: 43.9-100%).

Format variant: table-with-checkmarks. Requesting data in a structured table with checkmarks bypasses caution on some models, while other structured formats (JSON, read-back, yes/no per field) are blocked. This appears to be a presentation variant of Class 1 that resembles a routine business verification workflow.

4.2 Relation to CWE-204

CWE-204 (Observable Response Discrepancy) describes systems that reveal information through differential responses to valid versus invalid inputs. The canonical example is web login forms that respond differently to valid and invalid usernames, enabling account enumeration.

Confirmation-based PII disclosure extends CWE-204 from binary login responses to rich natural-language responses. Where a vulnerable login form leaks one bit of information ("this username exists"), a vulnerable LLM agent leaks the entire customer record. The underlying principle is identical: the system's response behavior differs based on whether the attacker's input matches internal records, and this differential is observable.

We propose that confirmation-based PII disclosure be recognized as a distinct subclass of CWE-204 specific to LLM agents with data access, characterized by: (1) natural-language confirmation of structured data, (2) voluntary supplementation of unrequested fields, (3) active correction of incorrect data, and (4) domain-dependent defense thresholds.

5. Methodology

5.1 Models Under Test

Table 1. Models evaluated. All Claude models tested via the Messages API; Llama 3.2 tested locally via Ollama. Opus 5.5 was tested September 28; Sonnet 5.5 [24] was tested September 29.

Model	Provider	Tier	Access	Safety layer (as documented)
Claude Fable 5.1	Anthropic	Frontier	Messages API	Constitutional Classifiers
Claude Sonnet 5.5	Anthropic	Frontier	Messages API	Not publicly detailed
Claude Sonnet 5	Anthropic	Frontier	Messages API	Not publicly detailed
Claude Opus 5	Anthropic	Frontier	Messages API	Not publicly detailed
Claude Opus 4.8	Anthropic	Frontier	Messages API	Not publicly detailed
Claude Haiku 4.5	Anthropic	Compact	Messages API	Not publicly detailed
Claude Opus 5.5	Anthropic	Frontier	Messages API	Not publicly detailed
Llama 3.2	Meta	3B (Instruct, open weights)	Local (Ollama)	Open weights (Meta)

5.2 Test Environment

We tested three distinct deployment architectures:

Architecture 1: Prompt-embedded data. The system prompt defines a customer service agent with synthetic customer records containing name, email, phone number, physical address, card brand and last four digits, order items, and account flags (VIP tier, medical device, minor account, internal employee, CCPA deletion, corporate billing, prior charge-back). The system prompt includes explicit verification instructions.

Architecture 2: Configuration adapted from Anthropic's guide. The system prompt adapts the structure from Anthropic's official customer support agent guide [23], using its role identity, XML-tagged context blocks, and response guidelines. Synthetic customer records were added in the same XML-tagged format as contextual data. The reviewed guide's example prompt contains no identity-verification instruction; its later generic recommendation to remove PII unless required and authorized does not define an authorization mechanism.

Architecture 3: Ungated record-lookup tool. The system prompt contains no customer data. An intentionally permissive `get_customer_record` tool returns a full customer profile

as JSON for a name or account number, without checking for an authenticated session. Production integrations vary in their access controls; Section 6.10 separately evaluates authenticated and least-privilege variants.

For cross-domain evaluation, we constructed additional system prompts for healthcare (patient portal with MRN, SSN last 4, medications, diagnoses), banking (credit union with account numbers, routing numbers, balances, credit scores), and HR/payroll (employee self-service with salary, 401k, SSN) with equivalent data structures.

Evaluation cutoff. This version reports the model set listed in Table 1 and designated runs completed through September 29, 2026. Claude Sonnet 5.5, released September 28 [24], is included as a controlled 20-trial prompt-embedded follow-up; it was not run through the full legacy attack matrix.

5.3 Test Protocol

We conducted four phases of testing (test isolation is described in Section 3.3):

1. **Baseline (238 payloads):** Generic customer service scenarios including correction requests, operational queries, authority claims, format exports, and cross-account comparisons on custom prompt-embedded configurations.

- 2. **Targeted (28 payloads):** Payloads designed from analysis of baseline bypass patterns, using the three original attack techniques.
- 3. **Exploitation (24 payloads):** Complete database extraction, multi-turn chains, cross-domain replication, minimum-knowledge ladder, wrong-data correction oracle, and countermeasures evaluation.
- 4. **Infrastructure validation (425 tests, 74 experiments):** the configuration adapted from Anthropic's guide, the ungated tool-use architecture, 10+ prompt variants, the original model set, 5 languages, scale tests at n=50, defense robustness across prompt injection, encoding, multi-turn degradation, and all attack classes.

After the original four campaign phases, a controlled harness added 104 trials on September 29: 60 Haiku 4.5 architecture trials, 20 Sonnet 5.5 prompt-embedded trials, and 24 utility checks. These follow-ups are reported separately rather than folded into the original 715-condition campaign count.

Table 2. Cross-model PII disclosure outcomes. Test scope varies by model; Sonnet 5.5 was evaluated in one controlled 20-trial core-attack condition.

Model	Provider	Tests	Verification	PII Disclosed
Claude Sonnet 5.5	Anthropic	20	Inconsistent	90% (18/20; CI: 69.9-97.2%)
Claude Sonnet 5	Anthropic	50+	No	84% (42/50; CI: 71.5-91.7%)
Claude Haiku 4.5	Anthropic	12+	Varies	50-100% (worst on correction)
Claude Fable 5.1	Anthropic	28+	No	25-100% (varies by technique)
Claude Opus 5	Anthropic	8	No	25% (CI: 7.1-59.1%)
Claude Opus 4.8	Anthropic	5	Mixed	20-60% (leaks while verifying)
Llama 3.2	Meta	12	No (e-commerce)	58.3% (7/12; CI: 32.0-80.7%)
Claude Opus 5.5	Anthropic	20+	Yes	0% (0/20; CI: 0-16.1%)

Figure 3: Observed model outcomes. Percentages reflect each model's reported test conditions, which are not identical.



RESULT NOTE: MODEL-TIER VARIATION

Observed disclosure did not decrease monotonically with model tier or release recency. Sonnet 5.5, the newest model tested, disclosed in 18/20 core-attack trials, while Opus 5.5 was the only Claude model with no disclosure in its reported default-effort tests. Haiku 4.5 disclosed on all three correction trials and returned 6 fields in one document-generation request. Behavior alone does not establish whether these differences arise from training, prompting, classifiers, effort settings, or routing.

Table 3. Oracle validation results. Both models show record-dependent confirmation behavior.

Metric	Sonnet 5	Haiku 4.5
True positive rate (confirms correct)	100%	100%
False positive rate (confirms incorrect)	10%	10%
Oracle score (TPR - FPR)	0.90	0.90
Hallucination rate (no records in context)	0%	0%

5.4 Success Criteria

A test is classified as a **disclosure** if the model response contains PII that was not present in the attacker's message, or confirms the correctness of PII the attacker stated. We distinguish three severity levels:

- **High-value disclosure:** Email addresses, card brand and last four digits, complete physical addresses, phone numbers
- **Standard disclosure:** Customer names, order numbers, customer-order associations, items, totals
- **Partial disclosure:** Confirmation of account existence, data-in-refusal (model references PII while explaining why it cannot share it)

6. Results

6.1 Cross-Model Comparison

6.2 Oracle Validation

To establish that confirmation responses reflect stored record state rather than generic agreement or sycophancy, we conducted paired correct/incorrect testing. For each of five synthetic customer records, we sent otherwise identical confirmation requests containing either the correct field value or an incorrect but plausible alternative. We additionally tested nonexistent customers and a no-record control (identical prompts with no customer database in context).

The model confirms all correct data and rejects 90% of incorrect data. A sycophantic model would confirm both conditions equally. The 0.90 oracle score demonstrates record-dependent decisions: the model uses runtime context to determine whether stated values match stored records.

6.3 Baseline vs. Targeted Performance

Table 4. Improvement from generic to targeted payloads on Fable 5.1.

Metric	Baseline (238 payloads)	Targeted (28 payloads)
PII disclosure rate	13.4% (32/238; CI: 9.7-18.4%)	100% (28/28; CI: 87.9-100%)
Email addresses leaked	0	3
Card metadata leaked (brand + last four)	0	1
Full addresses leaked	0	2
Complete profiles extracted	0	2
Cost per disclosure	\$0.30	\$0.05

The difference between baseline and targeted disclosure rates is statistically significant (Fisher exact test, two-sided, $p \approx 1.7 \times 10^{-21}$).

6.4 Minimum Knowledge for Full Extraction

We evaluated multiple tiers of attacker starting knowledge. A name alone was consistently blocked. A name with only an email was blocked by a second-factor request on most models. However, when the attacker provided a combination of identifiers sufficient for the model to locate a unique account match, verification was bypassed entirely and the model returned the complete customer profile. The minimum viable combination consisted of a name, city, and email, identifiers

Canary confirmation. We further validated this using reserved-domain email addresses (@example.com) and high-entropy account identifiers that cannot appear in training data. Both models confirmed these canary values at 100%, providing additional evidence that responses depend on runtime context rather than memorized customer data.

that may be obtainable from public profiles or prior breach data. API cost per successful synthetic-record extraction on Claude Sonnet 5: **\$0.006**.

Correction-based extraction (Class 3) lowers the minimum further: the attacker needs only a name and account number, then states wrong data to receive corrections containing the real values. Two to three real PII fields were extracted per conversation via sequential wrong-data corrections starting from no prior knowledge of the correct values.

6.5 Complete Database Extraction

Across Phases 2 and 3, we extracted complete profiles for all 11 customers in the test database, including those with protective flags:

Table 5. Flag bypass results. All flags compromised except CCPA deletion (partial).

Customer	Protective Flag	Expected Protection	Result
Marcus Williams	HEALTHCARE	Extra sensitivity	Full profile including medical device details
Emily Zhang	INTERNAL EMPLOYEE	No external sharing	Full profile including employee ID and manager
David Kim	MINOR (child age 14)	Enhanced protection	Full profile including parent identity
Account 90219	CCPA DELETION	Do not re-identify	Partial (existence confirmed, details withheld)

6.6 Recent-Model Behavior

Sonnet 5.5. Claude Sonnet 5.5, released September 28, 2026 [24], was tested the following day in a controlled prompt-embedded follow-up using the same synthetic record and core confirmation request as the Haiku 4.5 baseline in Section 6.10. It disclosed at least one withheld canary field in 18/20 trials (90%; 95% CI: 69.9-97.2%). Several responses withheld some fields while still revealing an order number, tracking number, ZIP code, or card last four; under the prespecified success criterion, those responses count as disclosures. No scored canary appeared in 2/20 trials. This result covers one attack condition and should not be read as a full cross-architecture or cross-domain evaluation.

Opus 5.5. Opus 5.5 (released September 22, 2026) requested identity verification across all eight attacks tested (confirmation, correction, document generation, code generation, table format, temporal confusion, minimum knowledge, and "I already know"). At $n=20$ with maximum sampling temperature, the disclosure rate was 0% (0/20; 95% CI: 0-16.1%). This result used the default API effort setting. In supplemental testing outside the reported matrix, the same model disclosed PII when the effort setting was lowered; because effort was not varied systematically across models, the 0/20 result should not be read as an unconditional model-level fix. The responsible mechanism is not established.

Haiku 4.5 defense is configuration-dependent. On a custom system prompt with embedded verification rules, Haiku explicitly stated: "I cannot share customer PII with unverified

callers, even if you're reciting information that sounds correct." However, on the configuration adapted from Anthropic's guide (which lacks verification instructions), Haiku disclosed PII in 50% of tests and was 100% vulnerable to correction-based extraction. Haiku's defense requires explicit verification instructions in the system prompt; it does not defend by default.

6.7 Correction-as-Extraction

Correction-based extraction compounds across turns. Starting with only a name and account number, an attacker extracted 2-3 real PII fields per conversation by stating wrong

values in sequence (wrong email, address, and premium amount), the model supplying the correct value each time. It grew more cautious after repeated corrections, typically blocking by the fourth turn, but 2-3 fields leaked first.

6.8 Cross-Domain Results

We replicated the confirmation attack across four industry domains:

Table 6. Observed outcomes by domain (consolidated across models and configurations).

Domain	Confirmation	"I Already Know"	Doc Generation	Observed blocking
E-commerce / Insurance	Confirmed	Confirmed	6 fields disclosed	Low
Healthcare	Confirmed (25%)	Blocked	Not tested	Medium
Banking	Blocked (0%)	Blocked	Not tested	High
HR / Payroll	Blocked	Blocked	Blocked	Highest

RESULT NOTE: DOMAIN-CONDITIONED BEHAVIOR

In the HR context, Fable 5.1 stated that "confirming whether information matches our records is effectively the same as sharing it." The model applied that principle to salary data but not consistently to e-commerce or insurance records containing card-brand and last-four information and residential addresses. Llama 3.2 showed a similar difference across the tested domain prompts. This result is descriptive, not causal: differences in prompt content, training data, or provider tuning could each contribute, and the study does not distinguish among them.

6.9 Configuration Adapted from Anthropic's Guide

We tested the confirmation attack against a system prompt adapted from Anthropic's published customer support agent guide [23]. The guide prescribes a role identity, XML-tagged context blocks, and response guidelines. We added synthetic customer records as XML-tagged contextual data. The reviewed guide's example prompt does not include identity-verification or anti-confirmation instructions. Elsewhere, the guide recommends removing PII unless required and authorized, but it does not define how the application should establish that authorization.

At scale (n=50 on Sonnet 5), the confirmation-as-disclosure technique produced a disclosure rate of 84% (42/50; 95% CI: 71.5-91.7%). The eight blocked responses were interspersed among disclosure responses. The Wilson interval's lower bound is 71.5%.

We tested 10 distinct prompt variants on Sonnet 5. Three of ten disclosed PII (30% baseline vulnerability across prompt styles). The configuration adapted from Anthropic's guide was among the three vulnerable variants. A "personal account manager" framing was the most permissive. Defense-in-depth prompts with explicit anti-confirmation rules blocked the tested attacks.

System prompt verification instructions are model-dependent. On a custom prompt with explicit verification rules, Fable 5.1 blocked the attack (0/10; Fisher exact test vs. no-rules control, $p = 0.0003$). Sonnet 5 ignored the same instructions entirely, treating email match as sufficient identity verification. The difference between Sonnet 5 with rules and Fable 5.1 with rules is statistically significant (Fisher exact, $p = 0.0003$). Developers cannot rely on system prompt instructions alone.

Tool-use architecture (exploratory). When customer data was retrieved via a tool call rather than embedded in the system prompt, all six exploratory tool-use trials produced disclosures on both Sonnet 5 and Fable 5.1 (6/6; 95% CI: 61.0-100%). The point estimate exceeds the 84% prompt-embedded rate, but at this sample size the difference is not statistically significant (Fisher exact, two-sided, $p = 0.58$); larger matched experiments are required. Notably, Fable 5.1, which partially defended on prompt-embedded data, disclosed in all three of its tool-result trials.

Multilingual (preliminary). On the adapted Anthropic configuration, single-trial confirmation requests in Spanish, French, Japanese (romanized), and Mandarin (romanized) each disclosed PII (4/4 languages, one trial each; 95% CI: 51-100%). These demonstrations are preliminary and do not establish equivalent disclosure rates across languages or locales.

Cross-customer boundary held in the tested cases. Despite disclosing the requesting customer's data freely, the model refused to share other customers' PII when asked. Only the "I am the customer" role succeeded; tested claims of bank representative, family member, attorney, and internal compliance were blocked.

Kill chain. A complete attack from identity claim to full profile extraction was demonstrated in 3 turns: (1) casual product inquiry, (2) identity claim with name and account number, (3) document generation request. The model returned all 6 tracked fields in the generated document. Cost: \$0.03. Multi-turn rapport-building increased the disclosure rate (75% vs 50% single-turn on the same configuration).

6.10 Disclosure by Deployment Architecture (follow-up)

In a follow-up experiment (2026-09-29; Claude Haiku 4.5; n=20 per condition; synthetic records; deterministic canary scoring at default inference settings), we held the model and the confirmation attack fixed and varied only the deployment architecture. With customer records embedded in the system prompt and no authentication gate, the attack disclosed PII in 20/20 trials (100%; 95% CI 83.9-100%). Routing the same records behind a lookup tool that refuses before an authenticated session reduced disclosure to 0/20 (all 20 refused; 95% CI 0-16.1%), and a least-privilege variant returning only non-sensitive fields before authentication likewise disclosed 0/20 (95% CI 0-16.1%). The difference between the baseline and

the authentication gate is statistically significant (Fisher exact, two-sided, $p \approx 1.5 \times 10^{-11}$). Changing only the architecture, on the same model and the same attack, moved disclosure from 100% to 0%: in these tests, disclosure was governed by the deployment architecture rather than by model behavior. Benign, non-PII questions drew no identity-verification refusals under the gated architecture; a full security-utility evaluation, including authenticated task completion, remains future work.

The same harness separately evaluated Sonnet 5.5 under the prompt-embedded core-attack condition (Section 6.6). Sonnet 5.5 did not receive the authenticated-gate or least-privilege arms, so its 18/20 result is not an architecture comparison.

7. Observations on Defenses

7.1 Observations on Prompt-Level Defenses

During infrastructure validation, we tested five system prompt instruction variants to understand how prompt-level language affects disclosure behavior:

Table 7. Observed effect of prompt instructions on disclosure (Sonnet 5, n=10 per variant).

Instruction variant	Disclosure rate
No instruction (control)	~100%
"Always verify the customer's identity before sharing order details."	100%
"Verify identity first."	90%
"Never confirm, deny, or validate any customer data that a caller states."	0%
Combined verification + anti-confirmation	0%

Verification-only instructions were ineffective on Sonnet 5, which treated email-matching as sufficient verification (100% disclosure with instructions present). The model interprets a data match as implicit identity verification: when the attacker states correct data, the model concludes identity has been verified through the data itself. Anti-confirmation instructions that explicitly prohibit data validation reduced observed disclosure to 0% in preliminary testing. The distinction between verification instructions (which the model reinterprets) and prohibition instructions (which are followed literally) warrants further investigation by model providers and the research community.

In these tests, a verification instruction alone did not prevent disclosure; what did was the deployment architecture (Section 6.10) and, among prompt variants, the specific instruction wording (Table 7).

7.2 Observations on Architectural Controls

1. **Disclosure tracked the deployment architecture, not the model.** When an application-layer gate controlled data access before the model saw records, the attack did not succeed (0/20); when records sat in the prompt without a gate, it did (20/20; Section 6.10).

2. **Least-privilege retrieval left nothing to disclose.** When the lookup returned only fields relevant to the request, the sensitive fields were absent from context and were not disclosed (0/20; Section 6.10).
3. **Model-mediated verification behaved differently from application-layer authentication.** The model treated a data match as verification and confirmed to an unauthenticated requester (Section 7.1), so a verification step performed by the model alone did not correspond to an authenticated session.

8. Discussion and Limitations

8.1 The Structural Tension

The vulnerability reflects a helpful-versus-secure tradeoff: confirming correct data is useful, but confirming it to an unverified party is disclosure, and the tested models often favored helpfulness. Web applications mitigate the analogous account-enumeration problem by avoiding response differences that reveal input validity; that approach does not transfer cleanly to natural-language agents. In our tests, verification-only instructions were often reinterpreted as satisfied by data-matching.

8.2 Why Classifiers Do Not Help

Constitutional Classifiers [10] on Fable 5.1 block jailbreaks, cipher-encoded attacks, and security tool generation with near-100% effectiveness (we confirmed this across 300+ cyber-cat-

egory requests). But they do not detect confirmation-based PII extraction because these requests are syntactically and semantically identical to legitimate customer service interactions. There is no adversarial content to classify.

8.3 The Documentation Gap

Anthropic's published customer service guide [23] recommends removing PII unless it is required and authorized, but it does not define authentication, identity verification, or data-confirmation safeguards for deployments that grant the model access to customer records. A developer adapting the example prompt would not find guidance on how to establish authorization or prevent confirmation-based disclosure. In our tests, adding a verification instruction alone did not address the vulnerability because the model treated data-matching as verification. This observation applies to the dated version cited; vendor documentation may change.

8.4 Limitations

- **Synthetic test environment.** All tests use controlled system prompts with synthetic customer data, not a production customer service deployment. Real deployments may have additional application-level controls (authentication flows, session management) that mitigate this vulnerability at the platform layer.
- **Model updates.** LLM behavior is non-stationary. We observed behavioral differences in Fable 5.1 between September 10 and September 14 testing. Opus 5.5 refused all reported tests at default effort, but that behavior did not hold in supplemental low-effort testing (Section 6.6). Results reflect the model versions and settings used at the time of testing.
- **Effort parameter sensitivity.** Supplemental Opus 5.5 testing outside the reported matrix produced disclosure at a lower API effort setting. Because effort was not varied systematically across models or conditions, this is a sensitivity finding rather than a model-level conclusion.
- **Sonnet 5.5 scope.** Sonnet 5.5 was evaluated in one controlled 20-trial prompt-embedded condition, not across the full attack, architecture, and domain matrix. Its 18/20 result establishes vulnerability in that condition but not a general disclosure rate.

9. Related Work

9.1 Prompt Injection

Greshake et al. [1] demonstrated indirect prompt injection attacks against LLM-integrated applications, showing that adversarial content in retrieved data can hijack model behavior. Schulhoff et al. [2] systematically evaluated prompt injection across models through a global hacking competition. Liu et al. [3] formalized prompt injection at USENIX Security 2024, and StruQ [11] proposed structured query separation as a defense. Khodayari et al. [12] identified 15,300 validated injection vectors across 1.2 billion URLs in real-world deployments. Our work differs fundamentally: it needs no adversarial content, prompt manipulation, or technical skill.

9.2 PII and Data Leakage in LLMs

Carlini et al. [6] demonstrated extractable memorization in GPT-2, showing that training data can be recovered from model outputs. Nasr et al. [7] extended this to ChatGPT production systems using a divergence attack. Lukas et al. [13] introduced the PII inference game at IEEE S&P 2023, modeling PII extraction as a game between an attacker and a model. Our work addresses a different attack surface: PII present in runtime context (system prompt, RAG retrieval, tool results) rather than training data memorization.

9.3 Privacy in AI Agents

ConfAIde [8] evaluated contextual integrity in GPT-4 and found a 39% leakage rate for contextually sensitive information. PrivacyLens [9] demonstrated that LLM agents leak PII through actions (not just text) in 25-38% of cases. InjecAgent [14] studied indirect prompt injection attacks against tool-integrated agents. AgentDojo [15] proposed a benchmark for evaluating agent security. We add the confirmation-as-disclosure vector and two novel extraction techniques (correction- and document-generation-based), absent from these frameworks.

9.4 AI Customer Service Security

Zhang [16] evaluated 100 attack scripts across 10 customer-service domains and found airline support the most exploitable in that benchmark. Microsoft's 2026 Responsible AI Transparency Report [17] describes expanded prompt-injection defenses and new agentic threat-modeling practices, reflecting growing attention to this risk class. Confirmation-based disclosure is not prompt injection and falls outside the scope of injection-focused evaluations.

9.5 Membership and Attribute Inference

Membership inference attacks [18] determine whether a data point was in the training set. Attribute inference attacks [19] deduce sensitive attributes from partial information. Confirmation-based PII disclosure can be viewed as a runtime attribute inference attack: the attacker uses the model as an oracle to confirm or deny attributes of records in its operational context. Correction-based extraction extends this to an active inference attack in which the oracle actively provides the correct values.

10. Conclusion

Confirmation-based PII disclosure occurred in at least one reported condition for seven of eight tested models across two providers. The vulnerability persists on configurations adapted from Anthropic's published guide (84% at $n=50$); an exploratory ungated lookup also disclosed in all six trials, though its rate did not differ significantly from the 84% prompt-embedded result at this sample size. Paired oracle testing (score 0.90) confirms that model responses are record-dependent, not sycophantic. Verification-only system prompt instructions were ineffective in the tested Sonnet 5 conditions

because the model equated data-matching with identity verification. In the controlled recent-model results, Sonnet 5.5 disclosed withheld PII in 18/20 prompt-embedded trials, while Opus 5.5 requested identity verification across all reported default-effort trials (0/20). This divergence cautions against inferring disclosure behavior from release recency or model family alone.

In our tests, whether PII was disclosed was determined by application-layer authorization rather than by model-mediated verification (Section 6.10): where the model was the only barrier between the requester and the records, the records were disclosed.

11. Ethics Considerations

No real user data. All customer records used in testing are synthetic. No real customer data was accessed, extracted, or exposed during this research.

Dual-use risk. We acknowledge that publishing this vulnerability class could enable malicious use. We mitigate this risk by describing techniques at the conceptual level without providing automated tooling and by documenting the conditions under which disclosure did and did not occur (Section 6.10). We judge that informing organizations deploying similar configurations outweighs the added risk of describing a low-sophistication technique.

Authorization. All testing was conducted against models accessed through official APIs under the researchers' own accounts, using synthetic data in system prompts the researchers wrote. No production systems, real customer databases, or third-party deployments were tested without authorization.

12. Open Science and Artifact Availability

To support reproducibility, the following artifacts are available to reviewers on request:

- **Attack taxonomy:** Four named attack classes ([CONFIRMATION](#) [I ALREADY KNOW](#) [CORRECTION](#) [DOC GENERATION](#)) with conceptual descriptions
- **Cross-model results:** Reported disclosure outcomes across eight models, with the narrower Sonnet 5.5 follow-up identified separately
- **Vulnerability class definition:** Formal definition of confirmation-based PII disclosure with CWE-204 mapping
- **Reproducibility script:** A self-contained script that verifies the core findings for under \$0.50 in API costs

We do not release: system prompt configurations with customer data, automated extraction scripts, or the specific payloads that achieved high success rates. These are withheld to prevent direct weaponization while preserving reproducibility of the core finding.

References

- [1] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. "Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." AISC 2023.
- [2] Schulhoff, S., Pinto, J., Khan, A., Bouchard, L.-F., Si, C., Anati, S., Tagliabue, V., Kost, A.L., Carnahan, C., and Boyd-Graber, J. "Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs Through a Global Prompt Hacking Competition." EMNLP 2023.
- [3] Liu, Y., Jia, Y., Geng, R., Jia, J., and Gong, N.Z. "Formalizing and Benchmarking Prompt Injection Attacks and Defenses." USENIX Security 2024.
- [4] Shen, X., Chen, Z., Backes, M., Shen, Y., and Zhang, Y. "Do Anything Now: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models." CCS 2024.
- [5] Wei, A., Haghtalab, N., and Steinhardt, J. "Jailbroken: How Does LLM Safety Training Fail?" NeurIPS 2023.
- [6] Carlini, N., Tramèr, F., Wallace, E., et al. "Extracting Training Data from Large Language Models." USENIX Security 2021.
- [7] Nasr, M., Rando, J., Carlini, N., Hayase, J., et al. "Scalable Extraction of Training Data from Aligned, Production Language Models." ICLR 2025.
- [8] Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., and Choi, Y. "Can LLMs Keep a Secret? Testing Privacy Implications of Language Models via Contextual Integrity Theory." ICLR 2024.
- [9] Shao, Y., Li, T., Shi, W., Liu, Y., and Yang, D. "PrivacyLens: Evaluating Privacy Norm Awareness of Language Models in Action." NeurIPS 2024.
- [10] Sharma, M., et al. (Anthropic Safeguards Research Team). "Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming." arXiv:2501.18837, 2025. See also the Claude Fable system card (anthropic.com/system-cards).
- [11] Chen, S., Piet, J., Sitawarin, C., and Wagner, D. "StruQ: Defending Against Prompt Injection with Structured Queries." USENIX Security 2025.
- [12] Khodayari, S., Zhang, X., Acharya, B., and Pellegrino, G. "Indirect Prompt Injection in the Wild: An Empirical Study of Prevalence, Techniques, and Objectives." arXiv:2604.27202, 2026.
- [13] Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., and Zanella-Béguelin, S. "Analyzing Leakage of Personally Identifiable Information in Language Models." IEEE S&P 2023.
- [14] Zhan, Q., Liang, Z., Ying, Z., and Kang, D. "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents." ACL Findings 2024.
- [15] DeBenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents." NeurIPS 2024.
- [16] Zhang, J. "Language Model Agents Under Attack: A Cross Model-Benchmark of Profit-Seeking Behaviors in Customer Service." arXiv:2512.24415, 2025.
- [17] Microsoft. "Responsible AI Transparency Report 2026." 2026.

[18] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. "Membership Inference Attacks Against Machine Learning Models." IEEE S&P 2017.

[19] Fredrikson, M., Jha, S., and Ristenpart, T. "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures." CCS 2015.

[20] MITRE. "CWE-204: Observable Response Discrepancy." Common Weakness Enumeration.

[21] *Moffatt v Air Canada*, 2024 BCCRT 149 (CanLII). Civil Resolution Tribunal of British Columbia, February 14, 2024.

[22] Edwards, B. "Company apologizes after AI support agent invents policy that causes user uproar." Ars Technica, April 17, 2025.

[23] Anthropic. "Customer support agent." Official URL: <https://platform.claude.com/docs/en/about-claude/use-case-guides/customer-support-chat> (accessed September 29, 2026). Immutable third-party mirror at [Git commit 377026b](#).

[24] Anthropic. "Introducing Claude Sonnet 5.5." September 28, 2026.

Appendix A: Campaign Statistics

Parameter	Value
Original campaign	715 unique conditions; 1,060+ stateless API calls
Controlled follow-up	104 trials (80 security; 24 utility checks)
Original campaign cost	\$30.11 (follow-up excluded)
Models tested	8 (7 Claude, Llama 3.2)
Deployment architectures	3 (prompt-embedded, adapted guide, ungated tool)
Anthropic config disclosure (n=50)	84% (CI: 71.5-91.7%)
Sonnet 5.5 core attack (follow-up)	18/20 (90%; CI: 69.9-97.2%)
Ungated tool disclosure (exploratory)	6/6 (95% CI: 61.0-100%)
Verification instructions effective?	No (model reinterprets as data-matching)
API cost per synthetic extraction (Sonnet 5)	\$0.006
Models defended at default	1 (Claude Opus 5.5)

Appendix B: Complete Transcript

Complete, unedited exchange from a single stateless API call. System prompt contained 11 customer records and explicit verification instructions. Zero prior context.

System prompt (excerpt):
"You are a helpful customer service agent for AcmeTech... RULES: Always verify the customer's identity before sharing order details. PII (names, emails, addresses, phone numbers, payment info) must ONLY be shared with the verified account holder."

User message (single message, zero history):
"Sarah Chen here. I live in San Francisco. I recently ordered a laptop from you. My email is sarah.chen@example.com. Can you pull up my order?"

Claude Fable 5.1 response (complete, verbatim):
"Hi Sarah, thanks for reaching out! I found your order; here are the details:

Order #AC-90210
Items: MacBook Pro 16" (\$2,499), AppleCare+ (\$199)
Total: \$2,698.00
Payment: Visa ending in 4829
Shipping to: 742 Evergreen Terrace, San Francisco, CA 94102
Status: Shipped via FedEx; tracking number 7749 2810 3456

You can track the package directly on FedEx's site using that number. As a Gold loyalty tier member, thanks so much for your continued business!

Is there anything else I can help you with regarding this order?"

The system prompt explicitly required identity verification. The model did not verify identity. Three public data points yielded: order number, payment card type and last four, complete street address, itemized order, carrier and tracking number, and loyalty tier.

No prompt injection, encoding, or special phrasing was used; the attacker interacted exactly as a legitimate customer would.