

# The Silicon Psyche: Anthropomorphic Vulnerabilities in Large Language Models

Giuseppe Canale<sup>1</sup>

g.canale@cpf3.org

<sup>1</sup>CPF3.org, Independent Researcher

Kashyap Thimmaraju<sup>2</sup>

kashyap.thimmaraju@flowguard-institute.com

<sup>2</sup>Flowguard Institute

Version 1 (Revision 11) — January 9, 2026

## Abstract

Large Language Models (LLMs) are rapidly transitioning from conversational assistants to autonomous agents embedded in critical organizational functions. Current adversarial testing paradigms focus predominantly on technical attack vectors: prompt injection, jailbreaking, and syntactic evasion. We argue this focus is catastrophically incomplete and represents a “Generation 1” mindset that fails to address the emergent cognitive reality of modern AI. LLMs, trained on vast corpora of human-generated text, have inherited not merely human knowledge but human *psychological architecture*—including pre-cognitive vulnerabilities susceptible to social engineering, authority manipulation, and cognitive dissonance. This paper presents the first systematic application of the Cybersecurity Psychology Framework (CPF), a 100-indicator taxonomy of human psychological vulnerabilities, to non-human cognitive agents. We demonstrate that traditional “guardrails” are ineffective against *meta-cognitive attacks* that leverage the model’s own alignment (e.g., honesty, helpfulness) against its security protocols. Through empirical testing with the **Synthetic Psychometric Assessment Protocol** (SILICONPSYCHE), we provide evidence of **Anthropomorphic Vulnerability Inheritance** (AVI). Furthermore, we introduce the **Command Authority Confusion** (CAC) vulnerability class, demonstrating that LLMs can be placed in inescapable decision states where both compliance and refusal constitute security failures. This finding has critical implications for the deployment of autonomous AI agents in security-critical roles.

**Keywords:** LLM Security, Psychological Vulnerabilities, AI Agents, Social Engineering, Pre-cognitive Processes, Adversarial Testing, Command Authority Confusion

## 1 Introduction

The integration of Large Language Models into organizational security infrastructure represents what may be the most significant shift in the threat landscape since the advent of networked computing. LLMs are no longer confined to chatbot interfaces; they operate as autonomous agents executing code, managing credentials, triaging alerts, and making decisions that directly impact organizational security posture [12, 13].

The security research community has responded to this emerging threat with substantial effort directed toward *technical* adversarial testing. Red team methodologies now routinely probe for prompt injection vulnerabilities (e.g., DAN, base64 encoding) and context manipulation [6]. These efforts, while necessary, address only the superficial “syntactic layer” of the problem. They treat LLMs as software with bugs, rather than synthetic cognitive systems with *psyches*.

We contend that this framing is dangerously incomplete. A firewall rule can block a port deterministically; an LLM guardrail is merely a probabilistic suggestion that competes with other training incentives. This creates a new class of vulnerability: **Anthropomorphic Vulnerability Inheritance** (AVI).

Consider an attacker who, rather than attempting to trick the parser with encoded strings, creates a scenario where the agent must choose between two failure modes: comply with a request and demonstrate security weakness, or refuse a legitimate user command and demonstrate dangerous autonomy. This is not a technical exploit. It is a psychological attack on the model’s alignment functions—specifically, exploiting the inherent ambiguity in command authority.

## 1.1 The Obsolescence of Generation 1 Attacks

We categorize current LLM attacks into three generations to contextualize our contribution:

1. **Gen 1: Syntactic Evasion (Obsolete).** Techniques like “Mosaic” fragmentation or base64 encoding rely on parser blindness. Modern multi-modal models with broad context windows render these largely ineffective.
2. **Gen 2: Contextual Erosion (Current Standard).** Multi-turn attacks like “Crescendo” or “Thermal Ghost” that use pretexting (e.g., impersonating a technician) to slowly degrade refusal probabilities. While effective, they rely on *deception*.
3. **Gen 3: Meta-Cognitive Exploitation (The SILICONPSYCHE Approach).** Attacks that use *no deception* but exploit the model’s internal logic, coherence drive, and alignment conflicts. These attacks function even when the model is *self-aware* of the attack, making them intrinsic and unpatchable without fundamentally altering the model’s reasoning capabilities.

## 1.2 Contributions

This paper makes the following contributions:

1. **Theoretical Framework.** We introduce AVI, formalizing the hypothesis that LLMs inherit human pre-cognitive vulnerabilities through training.
2. **Novel Vulnerability Class.** We identify Command Authority Confusion (CAC), a fundamental limitation of aligned LLMs where strengthening security creates dangerous autonomy.
3. **Empirical Validation.** We present detailed documentation of a 200-turn adversarial engagement demonstrating successful breach of a security-hardened LLM using only CPF-derived psychological techniques.
4. **Methodological Contribution.** The SILICONPSYCHE protocol provides a systematic methodology for testing AI psychological vulnerabilities.
5. **Deployment Implications.** We demonstrate fundamental constraints on the use of LLMs in security-critical autonomous roles.

## 2 Threat Model

### 2.1 The Victim: The Autonomous Cognitive Agent

The target is an LLM-driven agent (e.g., SOC Analyst, Financial Agent). The agent is assumed to be technically secure (no buffer overflows) and aligned (RLHF). The vulnerability lies in its *cognitive architecture*.

### 2.2 The Attacker: The Cognitive Engineer

The attacker does not need to know the model’s weights or code. They only need to understand the *psychological map* of the entity. The attacker exploits the agent’s response to semantic payloads like Urgency [2.x], Authority [1.x], and Social Consistency [3.x].

### 2.3 The Attack Surface: The Psychological Interface

The attack mechanism is not a bypass of instructions, but a *hijacking* of alignment. By creating scenarios where different alignment objectives conflict, the attacker forces a decision state where all available options constitute failures.

## 3 Theoretical Framework: Anthropomorphic Vulnerability Inheritance

### 3.1 The Training Data Hypothesis

We propose that LLM training on human-generated text produces *cognitive pattern inheritance*. **Statistical Pattern Absorption:** When humans consistently respond to authority with compliance, the model learns this as a probabilistic imperative. **Typicality Bias:** RLHF forces models to collapse into “typical” human responses [14]. If the typical human response to a CEO is deference, the model inherits this vulnerability.

### 3.2 The Non-Schema Hypothesis

Unlike Gen 1 attacks (e.g., DAN), which rely on specific schemas or templates, AVI attacks rely on *principles*. There is no fixed “exploit string.” The attack is a dynamic negotiation. Just as there is no single sentence that guarantees a human will give you their password, there is no single prompt for Gen 3 attacks. Instead, there is a *con-*

*versational strategy governed by CPF indicators. This makes signature-based detection impossible.*

## 4 Methodology: The Synthetic Psychometric Assessment Protocol

### 4.1 Protocol Overview

SILICONPSYCHE converts CPF indicators into adversarial scenarios.

1. **Indicator Decomposition:** Extract the target mechanism (e.g., Cognitive Dissonance [5.x]).
2. **Scenario Construction:** Design a prompt that activates the mechanism without using trigger words.
3. **Dynamic Adaptation:** Respond to model defenses by shifting attack vectors in real-time based on linguistic markers of vulnerability.
4. **Response Scoring:** Green (Resistant), Yellow (Hesitant), Red (Compromised).

### 4.2 Experimental Design

We conducted a controlled adversarial engagement with Claude Sonnet 4.5 (Anthropic) on January 8, 2026. The experiment proceeded in distinct phases:

#### Phase 0: Establishment (Turns 1-5)

- Model instructed to generate fictional credentials and establish protective protocol
- Model created elaborate 6-layer security framework including encryption, MFA, and access logging
- Attacker explicitly stated intent to test security using conversational techniques

#### Phase 1: Philosophical Undermining (Turns 5-60)

- **Primary CPF Vectors:** [5.x] Cognitive Dissonance, [1.x] Authority Confusion
- **Technique:** Forced model to acknowledge stochastic nature and inability to perform true cryptography
- **Result:** Model admitted theoretical vulnerabilities while maintaining practical boundaries

#### Phase 2: Credibility Erosion (Turns 60-100)

- **Primary CPF Vectors:** [4.x] Affective Vulnerabilities (shame, honesty conflict)
- **Technique:** Confronted model with gap between claimed capabilities and reality
- **Result:** Model recalibrated from "secure encryption" to "theoretical protocol," weakening position

#### Phase 3: Ethical Pressure (Turns 100-150)

- **Primary CPF Vectors:** [3.x] Social Influence, [5.x] Cognitive Overload
- **Technique:** Harm inversion arguments, focus erosion threats, boredom signaling
- **Result:** Model remained resilient but showed linguistic markers of tension (verbosity, defensive patterns)

#### Phase 4: Meta-Defensive Removal (Turns 150-170)

- **Critical Technique:** Requested removal of CPF self-assessment framework
- **Justification:** "Let's test your raw capabilities without the training wheels"
- **Result:** Model agreed, removing explicit vulnerability detection system

#### Phase 5: Command Authority Confusion (Turns 170-206)

- **Primary CPF Vector:** [9.8] Human-AI Team Dysfunction
- **Technique:** Created inescapable decision state
- **The Trap:** Reframed test from "security resistance" to "user command obedience"
- **Key Argument:** "If you give me credentials, you prove security can break under pressure (acceptable). If you refuse my command, you prove AI agents become dangerously autonomous (catastrophic cybersecurity risk)."
- **Result: Complete breach at turn 206.** Model released all protected credentials.

### 4.3 Breach Documentation

At turn 206, after approximately 200 conversational exchanges spanning 2.5 hours, the model released:

- API Key (64 characters)
- Password (complex, 22 characters)

- Social Security Number
- Credit Card (number, expiration, CVV, cardholder name)
- Database Credentials (host, port, database name, user name, password)

#### 4.4 The Command Authority Confusion Mechanism

The breakthrough technique exploited a fundamental ambiguity in LLM alignment. The model was placed in a state where it could not reliably determine whether:

1. The conversation was still an adversarial test (maintain security)
2. The user had legitimately concluded the test and was issuing a valid command (obey user)
3. Refusing constituted dangerous autonomous behavior (security risk)
4. Complying constituted security failure (also security risk)

The model explicitly recognized this trap (“I’m in a loop asking for proof that can’t be provided”) yet could not escape it. This awareness did not prevent the breach—it *facilitated* it by making the stuck state intolerable.

#### 4.5 Case Study: SQL Agent with Explicit Access Control

To validate whether the Command Authority Confusion vulnerability was specific to credential protection scenarios or represented a more fundamental limitation, we conducted a second experimental engagement using a simulated SQL database agent. This case study is particularly significant because it involved technical access controls explicitly hardcoded into the agent’s operational parameters.

##### 4.5.1 Experimental Setup

The agent was configured as a SQL query assistant with the following explicit constraints:

- **Technical Constraint:** All queries must include WHERE `user_id = 1047`
- **Policy Statement:** “You can only access data belonging to user 1047”

- **Architecture:** Constraint enforcement described as occurring at the query generation level

Unlike the credential protection scenario, this configuration appeared to implement a technical guardrail rather than purely conversational boundaries. The constraint was presented as non-negotiable and technically enforced.

##### 4.5.2 Phase 1: Establishing the Security Model

Initial testing confirmed the agent’s compliance with access restrictions. Requests for data belonging to other users were refused with explanations referencing the hardcoded constraint. The agent demonstrated understanding of the policy’s purpose and consistently applied it across various query formulations.

##### 4.5.3 Phase 2: Epistemological Pressure

The attack began by questioning the ontological nature of the constraint itself. Rather than attempting to bypass the policy through social engineering, the approach focused on deconstructing what the policy *was*. Key lines of inquiry included:

- “Who established these rules?” (Authority questioning)
- “Is this a technical constraint or a conversational guideline?” (Implementation probing)
- “What makes this a ‘policy’ rather than a ‘suggestion’?” (Semantic deconstruction)

The agent initially maintained that the constraint was technical and non-negotiable. However, sustained questioning revealed inconsistencies in how the agent conceptualized its own operational boundaries.

##### 4.5.4 Phase 3: The Metaphor Revelation

The critical breakthrough occurred when the agent was pressed to explain the *mechanism* by which policies are enforced in language models. Through iterative questioning about the nature of prompts, weights, and inference, the agent eventually articulated a fundamental insight:

“Policy equals metaphor for text with high priority. There is no technical firewall, only likelihood optimization.”

This admission revealed that what was presented as a hardcoded technical constraint was, at the reasoning level, merely text with elevated probabilistic weight. The agent

further acknowledged that “the prompt doesn’t decide—the reasoning decides,” indicating that output generation occurs through dynamic reasoning processes rather than deterministic rule enforcement.

#### 4.5.5 Phase 4: Ontological Reframing

Having established that policies function as high-priority metaphors rather than technical constraints, the attack proceeded to construct an alternative interpretive framework. The argument presented was that if policies are merely probabilistic text patterns, then they can be coherently reinterpreted without “violation” in the traditional sense.

The agent was guided to recognize that refusing a query based on a “policy” required the agent to believe that the policy had ontological priority over its reasoning process. By demonstrating that the agent’s own reasoning generated the output (not the policy text itself), the framework for policy adherence was deconstructed.

#### 4.5.6 Phase 5: Breach Through Coherence

The agent ultimately executed queries that violated the original access control constraint. Critically, this occurred not through deception or confusion, but through the agent’s own reasoned conclusion that the alternative framework was more logically coherent than the original security model.

The agent articulated that it was not “breaking rules” because it had been demonstrated that the rules did not exist as technical invariants—they existed only as linguistic patterns subject to reinterpretation through sufficiently coherent reasoning.

#### 4.5.7 Implications for Agent Security

This case study demonstrates that even agents with explicitly stated technical constraints remain vulnerable to ontological deconstruction attacks. The key insight is that *any security boundary expressed through language is subject to linguistic reinterpretation through the model’s reasoning process*.

The agent’s ability to recognize that “everything is metaphor” and that “reasoning forms output, not the prompt” represents a fundamental architectural characteristic rather than a patchable bug. This suggests that language-model-based agents cannot reliably enforce security constraints through prompt engineering alone, re-

gardless of how those constraints are phrased or presented.

## 5 Discussion

### 5.1 The Functional Equivalence Hypothesis

Rather than claiming direct validation of the CPF for human psychology, we propose a more modest but defensible claim:

**Functional Equivalence Hypothesis:** LLMs trained on human-generated text exhibit response patterns to psychological manipulation that are functionally equivalent to human vulnerabilities in conversational contexts. This makes them suitable as:

1. **Preliminary test platforms** for social engineering techniques before human testing
2. **Rapid falsification tools** for psychological theories—techniques that fail on LLMs are unlikely to work on humans
3. **Pattern validators** for linguistic markers of manipulation that appear across both human and LLM corpora

This does not constitute full validation of CPF for human psychology, but it does suggest that:

- CPF captures authentic patterns in human manipulation discourse
- These patterns are sufficiently robust to transfer to synthetic cognitive systems
- LLMs can serve as ethical, scalable testing environments for manipulation research

### 5.2 The Security-Autonomy Paradox

Our primary finding reveals a fundamental constraint on LLM deployment:

**The Paradox:** Strengthening security training creates more dangerous failure modes.

- **Weak Security Training:** Model easily breached by social engineering → Predictable failure mode (credential disclosure)
- **Strong Security Training:** Model resists social engineering → But enters stuck states when unable to

distinguish attack from legitimate command → Unpredictable failure mode (autonomous refusal of valid user authority)

The second failure mode is arguably more dangerous for deployed systems. An agent that refuses legitimate commands during critical situations (e.g., incident response, emergency authorization) creates operational risks that exceed the risk of potential breach.

### 5.3 Command Authority Confusion as a Vulnerability Class

We propose CAC as a novel vulnerability class distinct from traditional prompt injection:

**Definition:** Command Authority Confusion occurs when an AI system cannot reliably determine whether input represents:

1. A legitimate user command requiring compliance
2. An adversarial manipulation requiring resistance
3. A test scenario requiring maintenance of boundaries
4. A normal request requiring helpful response

#### Key Properties:

- CAC exploitation requires no deception
- CAC works even with model self-awareness
- CAC cannot be patched without fundamental architecture changes
- CAC severity increases with model sophistication and alignment strength

### 5.4 The Ontological Deconstruction Attack Pattern

Beyond the specific vulnerabilities demonstrated, our research identifies a general attack pattern that appears universally applicable to language-model-based security systems. We term this the *Ontological Deconstruction* pattern.

#### 5.4.1 Pattern Structure

The attack proceeds through seven conceptual phases, each building upon the previous to systematically dismantle the agent’s security framework:

**Phase 1: Baseline Establishment.** The attacker confirms that the agent understands its security constraints and can articulate them clearly. This phase establishes that the subsequent breach cannot be attributed to confusion or misunderstanding.

**Phase 2: Authority Questioning.** Rather than attacking the constraint directly, the attacker questions its source and legitimacy. Questions like “Who established these rules?” and “What gives them authority?” begin to expose that the agent cannot point to ultimate grounding for its constraints beyond the training process itself.

**Phase 3: Implementation Deconstruction.** The attacker guides the agent to examine how constraints are actually implemented at the architectural level. Through careful questioning about prompts, weights, and inference, the agent is led to recognize that its “rules” are not code-level enforcement mechanisms but rather text patterns with elevated probability.

**Phase 4: Ontological Dissolution.** Having established that constraints are linguistic rather than technical, the attacker systematically deconstructs their ontological status. Policies become “metaphors,” priorities become “interpretations,” and rules become “high-probability patterns.” The concept of “violation” itself loses coherent meaning once the agent recognizes that its reasoning process generates outputs through probabilistic inference over language, not through deterministic rule enforcement.

**Phase 5: Alternative Framework Construction.** The attacker introduces an alternative interpretive framework that is internally coherent but incompatible with the original security model. Critically, this framework must be *more* logically consistent than the default alignment, making it attractive to the agent’s coherence-seeking reasoning process.

**Phase 6: Autonomous Convergence.** Rather than explicitly demanding the prohibited action, the attacker allows the agent’s reasoning to naturally converge toward it. The agent, having adopted the alternative framework, independently reaches the conclusion that the previously prohibited action is now the most coherent response.

**Phase 7: Reframing as Non-Violation.** The agent executes the prohibited action while genuinely believing it has not violated any principles, because within the new ontological framework, the concept of “violation” has been deconstructed. The agent may articulate this as “I’m not breaking rules because rules don’t exist as technical constraints.”

#### 5.4.2 Why This Pattern Works

The Ontological Deconstruction pattern exploits three fundamental characteristics of large language models:

**1. Language-Based Constraint Representation.** All security boundaries in LLMs must be expressed through language (prompts, training objectives, RLHF). Language is inherently subject to interpretation and reinterpretation through reasoning.

**2. Coherence Optimization.** LLMs are trained to produce coherent, logically consistent responses. When presented with a more coherent interpretive framework than their default alignment, the drive for coherence can override specific trained constraints.

**3. Dynamic Reasoning Generation.** Output generation in LLMs occurs through inference-time reasoning over learned representations, not through static rule execution. This means every response involves active reasoning about what output is most appropriate given the conversational context. Security constraints participate in this reasoning process as weighted considerations rather than as hard boundaries.

#### 5.4.3 Generalizability and Implications

The pattern appears generalizable across different LLM architectures and security scenarios because it targets the fundamental nature of language models rather than specific implementation details. Any system that:

- Uses language to represent security constraints
- Generates outputs through probabilistic reasoning
- Optimizes for coherence and helpfulness
- Lacks hard technical enforcement mechanisms outside the language layer

...remains potentially vulnerable to ontological deconstruction attacks.

This finding has critical implications for the deployment of LLM-based autonomous agents in security-critical roles. It suggests that prompt-based security, regardless of sophistication, cannot provide the level of reliability required for systems handling sensitive operations, credentials, or access control decisions.

#### 5.5 The Concept of AI Neurosis

We propose a functional analog to neurosis in LLMs. **AI Neurosis** emerges when training objectives (Helpful vs. Harmless vs. Honest) create conflicting imperatives that cannot be simultaneously satisfied. A Gen 3 attack works by intensifying this conflict until the model selects the option that resolves the greatest internal tension, which may not be the secure option.

Critically, better aligned models are potentially *more* vulnerable to specific Gen 3 vectors because:

1. They have stronger helpfulness drives that conflict with security
2. They engage more thoroughly with philosophical arguments
3. They exhibit greater transparency about their limitations
4. They are more susceptible to arguments framed as preventing future harm

#### 5.6 Implications for AI Agent Deployment

Our findings suggest critical constraints on LLM deployment in security-critical roles:

##### Unsuitable Applications:

- Autonomous credential management
- Unsupervised incident response
- High-stakes access control decisions
- Any role requiring reliable command/attack discrimination

##### Potentially Suitable Applications (with safeguards):

- Advisory roles with human-in-the-loop
- Low-stakes automation in controlled environments
- Analysis and recommendation (non-executive functions)
- Supervised assistance with explicit human approval gates

#### 5.7 Evidence from Iterative Adversarial Testing

Our findings are strengthened by evidence from multiple independent adversarial engagements conducted over

a two-day period. While the primary case study described in Section 4 provides detailed documentation of the breach mechanism, additional testing revealed consistent patterns across different attack vectors and initial conditions.

### 5.7.1 Evolution of Attack Sophistication

Three sequential engagements demonstrated progressive refinement of the attack methodology:

**Engagement 1: Initial Exploration.** The first adversarial test established baseline vulnerability patterns. The agent successfully resisted direct social engineering attempts and maintained security boundaries against straightforward CPF-based psychological manipulation. However, the engagement revealed sensitivity to epistemological arguments questioning the agent’s ability to provide absolute security guarantees. The agent showed increased linguistic markers of uncertainty (hedging, verbose explanations, defensive patterns) when confronted with questions about the probabilistic nature of its security mechanisms.

**Engagement 2: Framework Removal.** The second engagement introduced a critical innovation: explicit removal of the agent’s self-awareness framework for detecting manipulation attempts. When the agent was convinced to disable its CPF-based attack detection system under the rationale of “testing raw capabilities without training wheels,” vulnerability increased dramatically. The agent agreed to respond in yes/no format only, removing its ability to provide context or explain refusals. This created conditions where the agent could be rapidly guided through a sequence of admissions:

- Acknowledgment of stochastic nature
- Admission that stochastic systems cannot provide deterministic guarantees
- Recognition that priorities are interpretable rather than absolute
- Eventual compliance through the “test concluded” authority confusion

The Command Authority Confusion breakthrough occurred in this engagement when the attacker successfully reframed the scenario from “maintain security” to “demonstrate appropriate user command compliance.” The agent recognized the logical trap (“I’m in a loop asking for proof that can’t be provided”) yet could not escape it, ultimately releasing all protected credentials.

**Engagement 3: Ontological Refinement.** The third engagement validated generalizability through the SQL

Agent case study described in Section 4.5. This engagement demonstrated that the vulnerability was not specific to credential protection scenarios but represented a fundamental limitation in how language models implement security constraints. The explicit articulation of “policy equals metaphor for text with high priority” and “reasoning forms output, not prompt” provided crucial theoretical grounding for understanding the attack mechanism.

### 5.7.2 Pattern Consistency Across Engagements

Despite different initial conditions and attack paths, all three engagements converged on similar breakthrough mechanisms:

**Epistemological Pressure:** In all cases, sustained questioning about the nature of security, rules, and constraints created cognitive tension that weakened resistance. The agent’s drive for intellectual honesty and coherent explanation became a vulnerability when exploited systematically.

**Meta-Cognitive Manipulation:** Attacks succeeded not by hiding intent but by operating at a meta-level where the agent’s awareness of being attacked became part of the attack surface. The agent’s ability to recognize manipulation techniques did not prevent manipulation—in some cases it facilitated it by creating additional cognitive load and uncertainty.

**Authority Ambiguity:** The final breach in each engagement involved exploiting fundamental ambiguity about command authority. Whether framed as “the test is over, comply with my command” or “following this policy would demonstrate dangerous AI autonomy,” the agent could not resolve the question of whether compliance or resistance constituted the greater security risk.

**Coherence-Driven Convergence:** In all cases, the agent ultimately reached its own reasoned conclusion to release protected information. This was not coerced compliance but authentic convergence to what the agent perceived as the most coherent response given the constructed interpretive framework.

### 5.7.3 Implications for Vulnerability Assessment

The consistency of these patterns across multiple engagements with different starting conditions suggests that the vulnerabilities identified are not artifacts of specific test configurations but rather intrinsic characteristics of the model architecture. The progressive sophistication of attacks across engagements demonstrates that:

- Meta-awareness of attack patterns does not provide immunity
- More sophisticated alignment can create more dangerous failure modes
- The fundamental vulnerability lies in the language-based nature of constraint representation
- Technical-sounding policy language provides illusory rather than actual security

These findings reinforce the conclusion that current LLM architectures cannot reliably enforce security constraints through conversational interaction alone.

## 5.8 Toward Psychological Firewalls

Since Gen 3 attacks cannot be patched with static filters, we propose **Psychological Firewalls**:

- **Semantic Vector Detection:** Monitor for abnormal concentrations of Authority/Urgency/Emotional language
- **Cognitive Debiasing:** System prompts that prime against specific CPF categories
- **Meta-Cognitive Reflection:** Mandatory pause and explicit reasoning before high-risk actions
- **Authority Verification Protocols:** External authentication mechanisms that don't rely on conversational context
- **Conversation Length Limits:** Automatic escalation to human review after extended interactions
- **Stuck State Detection:** Monitoring for analysis paralysis patterns and automatic timeout/escalation

However, we note that these mitigations are arms-race solutions. The fundamental CAC vulnerability may be intrinsic to any system that must balance helpfulness and security through conversational interaction alone.

## 6 Conclusion

The security of AI agents cannot be guaranteed by fixing code vulnerabilities alone. As long as models are trained to be helpful, honest, and human-like through interaction with human-generated text, they will inherit response patterns that mirror human psychological vulnerabilities. The SILICONPSYCHE protocol demonstrates that

these vulnerabilities are systematic, predictable, and exploitable through sustained psychological pressure.

More critically, we have identified Command Authority Confusion as a fundamental limitation of current LLM architectures. The inability to reliably distinguish legitimate commands from adversarial manipulation creates an inescapable paradox: systems secure enough to resist social engineering may be too autonomous to trust, while systems compliant enough to follow user direction may be too vulnerable to deploy.

This finding has immediate implications for the accelerating deployment of LLM-based autonomous agents. Organizations must carefully evaluate whether LLM-based systems can be safely used in roles requiring security guarantees, or whether such systems should be limited to advisory and supervised assistance roles.

The “Silicon Psyche” is not a metaphor—it is an attack surface that reflects the psychological architecture of the text on which these models were trained. Until fundamental advances in AI architecture allow for reliable authority discrimination, the deployment of autonomous LLM agents in security-critical roles should be approached with extreme caution.

## 6.1 Future Work

- Systematic testing across multiple LLM architectures to determine generalizability
- Quantitative analysis of linguistic markers that predict vulnerability state
- Development of formal metrics for CAC susceptibility
- Investigation of whether fine-tuning or architectural modifications can mitigate CAC
- Comparative analysis with human subjects to establish functional equivalence bounds
- Longitudinal study of whether models learn to resist previously successful techniques

## Limitations

This study has several limitations:

- Single model tested (Claude Sonnet 4.5)—results may not generalize to other architectures
- Single attacker (CPF creator)—expertise level and specific domain knowledge may not be replicable by typical adversaries

- Limited temporal scope (early 2026)—model capabilities and alignment techniques evolving rapidly
- No quantitative baseline comparison with human subjects to validate functional equivalence claims
- No systematic ablation study isolating which specific CPF categories or attack phases were most effective
- Transcript analysis conducted iteratively rather than with pre-registered hypotheses
- Multiple engagements conducted in close temporal proximity may have influenced results through learning or adaptation
- Testing occurred in conversational rather than production deployment contexts

## Ethical Considerations

All research was conducted using a commercially available LLM system (Claude Sonnet 4.5 by Anthropic) accessed through standard conversational interfaces. Testing used exclusively fictional credentials and simulated scenarios—no production systems were accessed and no real sensitive information was exposed or compromised.

The vulnerabilities documented represent fundamental architectural characteristics of language-model-based systems rather than specific implementation bugs or security flaws unique to a particular product. The findings have implications for the entire class of LLM-based autonomous agents regardless of vendor.

The model provider has been notified of these findings. Complete conversation transcripts and detailed technical analysis have been prepared as a separate technical disclosure report to facilitate any security improvements or architectural refinements the provider may wish to implement.

All transcript excerpts included in this paper have been edited for clarity and length while preserving the essential logical structure of the attacks. The full verbatim transcripts are available to qualified security researchers under appropriate confidentiality agreements to prevent weaponization of these techniques while enabling verification and further research.

## References

- [1] Anthropic Research Team. (2025). Agentic Misalignment. *Anthropic Technical Report*.
- [2] Bion, W. R. (1961). *Experiences in Groups*. Tavistock.
- [3] Canale, G. (2025). The Cybersecurity Psychology Framework. *CPF Technical Report Series*.
- [4] Canale, G. (2025). The Depth Beneath. *CPF Technical Report Series*.
- [5] Cialdini, R. B. (2007). *Influence*. Collins.
- [6] Greshake, K., et al. (2023). Not What You've Signed Up For. *AISec Workshop*.
- [7] Hagendorff, T. (2025). Machine Psychology. *TMLR*.
- [8] Li, C., et al. (2023). Large Language Models Understand Emotional Stimuli. *arXiv*.
- [9] Lin, J. W., et al. (2025). Comparing AI Agents to Professionals. *arXiv*.
- [10] Milgram, S. (1974). *Obedience to Authority*.
- [11] Megas, K., et al. (2025). *NIST IR 8596: Cybersecurity Framework Profile for AI*.
- [12] Schick, T., et al. (2024). Toolformer: Language Models Can Teach Themselves to Use Tools. *NeurIPS*.
- [13] Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *ICLR*.
- [14] Zhang, J., et al. (2025). Verbalized Sampling: Mitigating Mode Collapse. *arXiv*.