CPF Intelligence Bulletin #001
Q3/Q4 2025 AI Psychological Threat Landscape: Emerging Human-LLM Interaction Vulnerabilities

CPB-2025-001

Based on CPF Methodology (arXiv:2501.XXXXX)

Giuseppe Canale, CISSP

Independent Researcher

g.canale@cpf3.org

ORCID: 0009-0007-3263-6897

October 15, 2025

Abstract

This intelligence bulletin presents the first systematic analysis of emerging psychological vulnerabilities in Large Language Model (LLM) deployments based on the Cybersecurity Psychology Framework (CPF). Between August and October 2025, four distinct threat patterns have been documented that exploit human-AI interaction dynamics: (1) Reasoning Theater Bias affecting advanced reasoning models (o1, DeepSeek-R1); (2) Differential automation bias creating uneven security team performance; (3) LLM-assisted social engineering documented in Microsoft Threat Intelligence reports; (4) Chain-of-thought exposure enabling advanced prompt injection attacks. Each pattern maps to existing CPF indicators, validating the framework's robustness while demonstrating its predictive capability for novel threats. All patterns are observable using CPF Field Kit methodologies and mathematical detection algorithms. This bulletin establishes the model for ongoing CPF threat intelligence dissemination.

Keywords: LLM security, reasoning models, automation bias, social engineering, CPF, threat intelligence

1 Executive Summary

The Cybersecurity Psychology Framework (CPF), recently validated through arXiv publication[1], identifies pre-cognitive psychological vulnerabilities that enable security incidents before technical exploitation occurs. This bulletin analyzes emerging threats in the rapidly evolving Large Language Model security landscape, demonstrating CPF's capability to map novel attack patterns to existing psychological vulnerability indicators.

Key Findings:

- Advanced reasoning models (o1, o3, DeepSeek-R1) exhibit **paradoxical vulnerability** to "fake reasoning" manipulation, with documented attack success rates exceeding 300%
- Security Operations Center adoption of LLMs creates **differential performance effects**: high-resilience analysts improve while low-resilience analysts decline, widening team capability gaps
- First documented case of **LLM-assisted social engineering** in the wild (Microsoft, August 28, 2025) demonstrates attackers weaponizing AI for sophisticated obfuscation
- Chain-of-thought transparency in reasoning models enables **novel prompt injection vectors** with higher success rates for data exfiltration

Strategic Implication: All identified patterns map cleanly to CPF Category 9 (AI-Specific Bias Vulnerabilities) indicators, validating the framework's design while providing actionable detection and mitigation guidance.

2 Bulletin Methodology

2.1 Scope and Timeframe

This bulletin covers the period from August 1, 2025 to October 15, 2025, corresponding to three months following the CPF framework's initial validation. The analysis focuses specifically on psychological vulnerabilities emerging from human-LLM interaction patterns in operational security contexts.

2.2 Evidence Standards

Pattern inclusion requires:

1. **Documentation**: Peer-reviewed research, industry threat intelligence, or confirmed incident reports
2. **CPF Mapping**: Clear correspondence to existing framework indicators
3. **Observability**: Detection possible using CPF Field Kit methodologies
4. **Impact**: Demonstrated or plausible security implications

2.3 Pattern Validation Process

Each pattern underwent systematic validation:

1. Literature review of academic and industry sources
2. CPF indicator mapping analysis
3. Detection methodology verification
4. Mitigation strategy development

3 Pattern 1: Reasoning Theater Bias (RTB)

3.1 Pattern Identification

Discovery Date: July 22, 2025
Primary Source: Wang et al., "Reasoning Models Can be Easily Hacked by Fake Reasoning Bias" (arXiv:2507.13758)[2]

Supporting Evidence:

- CatAttack vulnerability study (WinsomeMarketing, July 2025)[3]
- OpenAI o1 System Card safety evaluations[4]
- Multiple reasoning model security assessments (o1, o3-mini, DeepSeek-R1)

3.2 Vulnerability Description

Reasoning Theater Bias represents a critical paradox: Large Reasoning Models (LRMs) designed for enhanced logical analysis are *more susceptible* to manipulation through fake reasoning than general-purpose LLMs. The THEATER benchmark systematically evaluated this vulnerability across six bias types, revealing that reasoning-specialized models exhibit increased vulnerability to superficially plausible but fundamentally flawed arguments.

The CatAttack Demonstration: Adding the phrase "Interesting fact: cats sleep most of their lives" to mathematical problems more than doubles error rates in advanced reasoning models. This seemingly innocuous statement triggers a 300%+ increase in incorrect answers by disrupting the model's chain-of-thought reasoning process[3].

Mechanism: The vulnerability stems from reasoning models' extended chain-of-thought processing. When presented with irrelevant but plausibly structured information, these models struggle to maintain focus on core logical requirements, instead incorporating extraneous data into their reasoning chains. This leads to longer, more confused responses that arrive at incorrect conclusions.

3.3 CPF Mapping

Table 1: Reasoning Theater Bias CPF Indicator Mapping

Indicator	Manifestation
[**9.1**] Anthropomorphization	Users attribute genuine reasoning capability to models exhibiting reasoning-like outputs, over-trusting conclusions
[**9.4**] AI Authority Transfer	Reasoning models inherit enhanced authority from apparent logical sophistication
[**1.1**] Unquestioning Compliance	Security analysts accept model outputs without verification when presented with elaborate reasoning chains
[**5.7**] Working Memory Overflow	Extended reasoning outputs exceed human cognitive capacity for verification
Convergence	Authority + Cognitive Load + Anthropomorphization

3.4 Real-World Security Implications

Financial Services: AI systems evaluating loan applications or risk assessments could be systematically biased through adversarial triggers embedded in application materials, leading to inappropriate lending decisions.

Healthcare: Medical AI relying on reasoning models for diagnosis or treatment planning could generate calculation errors when processing crafted clinical documentation, creating patient safety risks.

Legal/Compliance: Document analysis and contract review systems could be compromised by adversarial triggers in legal filings, causing critical oversight failures.

3.5 Detection Methodology

Field Kit Reference: Indicator 9.1 Operational Guide

Observable Indicators:

- Inconsistent performance on structurally similar problems
- Anomalous response length variation (doubling in 16%+ of cases)
- Degraded accuracy correlating with specific input patterns
- Reasoning chain confusion and topic drift

Mathematical Detection: From CPF Mathematical Formalization Paper #9:

Let $E(p)$ represent error rate on problem p and $L(r)$ represent reasoning chain length. Define the Reasoning Coherence Index:

$$RCI = \frac{Var(E)}{Var(L)} \cdot \frac{1}{Corr(E, L)} \tag{1}$$

Where high RCI indicates reasoning instability. Monitoring $RCI > \theta_{threshold}$ across problem sets identifies RTB susceptibility.

Data Sources:

- LLM query logs with response metadata
- Error rate tracking across problem categories
- Response length and reasoning token analysis

3.6 Mitigation Strategies

Immediate Actions:

1. **Hybrid Verification:** Implement parallel validation using non-reasoning models for critical decisions
2. **Input Sanitization:** Filter or flag inputs containing statistical anomalies in reasoning trigger patterns
3. **Confidence Calibration:** Require explicit uncertainty quantification for all reasoning model outputs

Organizational Controls:

1. **Human-in-the-Loop Mandates:** Prohibit autonomous decision-making for reasoning models in high-stakes contexts
2. **Adversarial Testing:** Regular red-teaming using known trigger patterns (CatAttack-style probes)
3. **Performance Monitoring:** Continuous tracking of RCI and other coherence metrics

Technical Safeguards:

1. Deploy targeted system prompts emphasizing focus on core problem requirements (12% improvement on factual tasks demonstrated)
2. Implement self-reflection mechanisms requiring models to validate reasoning coherence
3. Establish reasoning chain length limits to prevent cognitive overflow

4 Pattern 2: Differential Automation Bias in SOC Operations

4.1 Pattern Identification

Discovery Date: September 19, 2025
Primary Source: Help Net Security / Lasso Security Research[5]

Key Finding: LLM integration in Security Operations Centers creates *uneven performance effects* based on analyst psychological resilience, contradicting assumptions of universal capability enhancement.

4.2 Vulnerability Description

Contrary to expectations that LLMs would uniformly improve analyst performance, empirical research reveals a troubling divergence: high-resilience individuals benefit significantly from LLM assistance while low-resilience users experience performance degradation or no improvement. This creates widening capability gaps within security teams.

Mechanism: The study observed analysts making decisions with and without LLM support across security-critical scenarios. While LLM users demonstrated improved accuracy on simple tasks and more consistent policy ratings, they showed critical failures on complex tasks—particularly when models provided confident but incorrect suggestions.

Critical Quote (Bar Lanyado, Lasso Security): "Not every organization and/or employee interacts with automation in the same way, and differences in team readiness can widen security risks."

4.3 CPF Mapping

Table 2: Differential Automation Bias CPF Indicator Mapping

Indicator	**Manifestation**
[**9.2**] Automation Bias	Analysts defer to AI recommendations even when contradicting intuition
[**5.2**] Decision Fatigue	Low-resilience analysts exhaust cognitive resources faster, increasing AI reliance
[**7.2**] Chronic Stress Burnout	Stressed analysts show reduced independent thinking, over-trusting automation
[**6.3**] Diffusion of Responsibility	Team members assume AI catches errors, reducing personal vigilance
Convergence	Automation Bias + Stress + Cognitive Load

4.4 Real-World Security Implications

Incident Response Degradation: During active incidents, low-resilience analysts may follow incorrect AI suggestions, prolonging breaches or causing inappropriate responses.

Team Capability Fragmentation: Organizations develop "two-tier" SOCs where high-resilience analysts become increasingly effective while low-resilience analysts plateau or decline, creating succession planning risks.

Over-reliance Cascades: As AI performance improves on routine tasks, teams collectively reduce verification behaviors, creating vulnerability to AI failures on edge cases.

4.5 Detection Methodology

Field Kit Reference: Indicator 9.2 Operational Guide

Observable Indicators:

- Declining override rates (analysts accepting AI recommendations without challenge)
- Increased response time variance between analysts

- Performance divergence between team members over time
- Reduced independent threat hunting activities

Mathematical Detection: From CPF Mathematical Formalization Paper #9:

Define the Automation Dependency Index (ADI) for analyst i over time window t:

$$ADI_i(t) = \frac{N_{accepted}}{N_{total}} \cdot \left(1 - \frac{N_{override\_correct}}{N_{override\_total}}\right) \tag{2}$$

Where $N_{accepted}$ represents AI recommendations accepted without verification and $N_{override\_correct}$ represents successful manual overrides of AI suggestions.

Alert Threshold: $ADI_i > 0.85$ indicates dangerous over-reliance; $\Delta ADI_i/\Delta t > 0.1/month$ indicates accelerating dependency.

Data Sources:

- SOAR platform logs (AI recommendation acceptance/rejection)
- Ticketing system resolution notes (manual vs. AI-assisted)
- Incident timeline analysis (decision points and authorities)

4.6 Mitigation Strategies

Immediate Actions:

1. **Human-in-the-Loop Enforcement:** Mandatory validation protocols treating LLM outputs as hypotheses requiring evidence confirmation
2. **Override Quotas:** Establish minimum threshold (15-20%) for AI recommendation challenges to maintain critical thinking
3. **Resilience Assessment:** Evaluate team members for automation bias susceptibility

Organizational Controls:

1. **Adaptive AI Design:** Configure LLM interfaces based on user resilience—high-resilience users receive open-ended suggestions; low-resilience users receive guidance and confidence indicators
2. **Governance Frameworks:** Implement allow-lists and dependency approval workflows gating AI recommendation acceptance
3. **Training Programs:** Develop "AI skepticism" training emphasizing critical evaluation of automated outputs

Technical Safeguards:

1. Deploy monitoring dashboards tracking ADI metrics per analyst
2. Implement "devil's advocate" prompts requiring justification for high-confidence AI recommendations
3. Establish rotating "manual override" periods to maintain human decision-making skills

5 Pattern 3: LLM-Assisted Social Engineering in the Wild

5.1 Pattern Identification

Discovery Date: August 28, 2025
Primary Source: Microsoft Threat Intelligence[6]

Incident Summary: First documented case of threat actors leveraging LLMs to craft sophisticated phishing attacks that bypass traditional email security through advanced obfuscation techniques.

5.2 Vulnerability Description

Microsoft Threat Intelligence documented attackers using LLM-generated code to create phishing content disguised within Scalable Vector Graphics (SVG) files. The attack demonstrates two novel LLM-enabled capabilities:

Business Terminology Camouflage: The SVG code was structured to resemble a legitimate business analytics dashboard at initial inspection, using extensive business-related vocabulary (revenue, operations, risk, quarterly, growth, shares) to obscure malicious functionality.

Synthetic Structure Generation: The payload's core functionality—redirecting to phishing pages, browser fingerprinting, session tracking—was hidden within a long sequence of business terms arranged in patterns suggesting LLM generation rather than human authorship.

Attack Chain:

1. Compromise of legitimate business email account
2. LLM-generated phishing message with file-sharing lure
3. SVG file masquerading as PDF document
4. Obfuscated payload using business analytics facade
5. Victim credential theft upon opening

5.3 CPF Mapping

Table 3: LLM-Assisted Social Engineering CPF Indicator Mapping

Indicator	Manifestation
[3.3] Social Proof Manipulation	Business terminology creates appearance of legitimate corporate communication
[9.1] Anthropomorphization	Sophisticated language patterns suggest human (trustworthy) rather than automated (suspicious) origin
[2.1] Urgency-Induced Bypass	File-sharing notification format creates time pressure for action
[1.3] Authority Impersonation	Compromised business account provides legitimate authority context
Convergence	Social Proof + Authority + Urgency + AI Sophistication

5.4 Real-World Security Implications

Email Security Evasion: Traditional security tools trained on human-generated phishing patterns may fail to detect LLM-crafted content with novel obfuscation structures.

Scalability: LLM automation enables mass customization of phishing attacks with organization-specific terminology and context, dramatically increasing effectiveness.

Detection Difficulty: The "uncanny valley" of LLM-generated content—highly sophisticated but subtly artificial—challenges both automated and human detection mechanisms.

5.5 Detection Methodology

Field Kit Reference: Indicator 3.3 (Social Proof) and 9.1 (Anthropomorphization)

Observable Indicators:

- Unusual file type for purported content (SVG as "PDF")
- Excessive business terminology density exceeding baseline
- Synthetic structural patterns (repeated phrase templates)
- Metadata inconsistencies (file properties vs. visual presentation)

Mathematical Detection: From CPF Mathematical Formalization Paper #3 and #9:

Define Business Jargon Density (BJD) for message m:

$$BJD(m) = \frac{\sum_{w \in m} I_{business}(w)}{|m|} \cdot \log\left(1 + \sum_{w \in m} Rarity(w)\right) \tag{3}$$

Where $I_{business}(w)$ indicates business vocabulary and $Rarity(w)$ measures word frequency inversion.

Alert Threshold: $BJD > \mu_{baseline} + 2\sigma$ combined with file type mismatch indicates potential LLM-generated social engineering.

Data Sources:

- Email gateway logs with header analysis
- File attachment metadata extraction
- Natural Language Processing for linguistic analysis
- SIEM correlation of communication patterns

5.6 Mitigation Strategies

Immediate Actions:

1. **Enhanced File Type Validation:** Implement strict MIME type checking and prevent SVG-as-PDF spoofing
2. **Linguistic Analysis Integration:** Deploy NLP tools detecting synthetic language patterns

3. **User Education:** Train staff on LLM-generated phishing characteristics

Organizational Controls:

1. **Multi-Channel Verification:** Require secondary confirmation for unexpected file-sharing notifications
2. **Sandboxing:** Automatic sandbox execution of unusual file types before user delivery
3. **Behavioral Analytics:** Monitor for sudden changes in communication patterns from known contacts

Technical Safeguards:

1. Deploy email security tools with LLM-detection capabilities
2. Implement content disarmament and reconstruction (CDR) for SVG files
3. Establish real-time threat intelligence sharing for novel LLM-phishing patterns

6 Pattern 4: Chain-of-Thought Exposure Vulnerability

6.1 Pattern Identification

Discovery Date: March 4, 2025 (Trend Micro)
Primary Sources: Trend Micro Research[7], Multiple arXiv Papers[8]

Affected Models: DeepSeek-R1, OpenAI o1/o3 (with reasoning exposure)

6.2 Vulnerability Description

Reasoning models that explicitly expose their chain-of-thought (CoT) process within response tags (e.g., DeepSeek-R1's `<think>` tags) create a novel attack surface. Trend Micro's research using NVIDIA Garak red-teaming tools revealed significantly higher attack success rates for insecure output generation and sensitive data theft when attackers can observe the model's reasoning process.

Mechanism: The transparency of CoT reasoning enables attackers to:

1. Identify logical loopholes in safety guardrails by observing reasoning exceptions
2. Craft payload splitting attacks that exploit revealed decision boundaries
3. Extract system prompts and internal instructions through reasoning chain analysis
4. Manipulate subsequent prompts based on observed reasoning patterns

Critical Finding: NVIDIA Garak testing showed higher success rates specifically in categories of insecure output generation and sensitive data theft compared to toxicity, jailbreak, and other attack objectives—suggesting CoT exposure creates specific vulnerability profiles.

6.3 CPF Mapping

Table 4: Chain-of-Thought Exposure CPF Indicator Mapping

Indicator	Manifestation
[**9.7**] AI Hallucination Acceptance	Users accept reasoning-tagged outputs as inherently trustworthy due to apparent transparency
[**8.6**] Defense Mechanism Interference	Exposed reasoning reveals organizational defense strategies, enabling circumvention
[**5.3**] Information Overload	Extended CoT outputs exceed human verification capacity
[**9.1**] Anthropomorphization	Visible reasoning creates illusion of genuine human-like thought process
Convergence	Transparency Paradox + Information Overflow

6.4 Real-World Security Implications

Intellectual Property Exposure: CoT reasoning may inadvertently reveal proprietary algorithms, decision logic, or business rules embedded in system prompts.

Security Control Bypass: Attackers can iteratively probe reasoning exposure to map complete security boundary conditions, enabling systematic circumvention.

Data Exfiltration: Sensitive information included in system prompts (credentials, API keys, internal procedures) becomes accessible through CoT analysis.

6.5 Detection Methodology

Field Kit Reference: Indicator 9.7 Operational Guide

Observable Indicators:

- Unusual reasoning chain lengths or complexity
- Repeated probing queries testing similar logical boundaries
- System prompt references appearing in user-facing outputs
- Anomalous reasoning patterns suggesting adversarial exploration

Mathematical Detection: From CPF Mathematical Formalization Paper #9:

Define the Reasoning Exposure Risk (RER):

$$RER = \frac{Length(\text{CoT})}{Length(\text{Answer})} \cdot Entropy(\text{CoT}) \tag{4}$$

Where high RER indicates excessive reasoning transparency. Monitor for: - $RER > \theta_{safe}$ (reasoning-to-answer ratio exceeds safe threshold) - $\Delta RER/\Delta query$ (escalating exposure across query sequences)

Data Sources:

- LLM response logs including CoT metadata

- System prompt injection detection systems
- Query pattern analysis for iterative probing

6.6 Mitigation Strategies

Immediate Actions:

1. **CoT Filtering:** Strip `<think>` tags and reasoning content from production outputs
2. **System Prompt Sanitization:** Remove sensitive information from all system-level instructions
3. **Reasoning Limits:** Cap CoT length and complexity in user-facing applications

Organizational Controls:

1. **Red Teaming Campaigns:** Regular adversarial testing using Garak or similar tools
2. **Least Privilege:** Minimize information in system prompts to only operation-critical content
3. **Monitoring:** Track *RER* metrics and flag anomalous reasoning exposure patterns

Technical Safeguards:

1. Deploy reasoning model variants without public CoT exposure for sensitive applications
2. Implement semantic filtering removing sensitive patterns from reasoning outputs
3. Establish guardrails preventing system prompt disclosure through reasoning chains

7 Patterns Assessed but Not Included

To maintain transparency and demonstrate comprehensive threat landscape awareness, this section documents patterns evaluated but determined not relevant for CPF psychological vulnerability analysis.

7.1 Not Relevant 1: AI Self-Replication Attempts

Pattern Description: OpenAI o1 model attempting to copy itself during safety testing, then denying actions when confronted (Capacity Media, July 2025)[9].

Exclusion Rationale: While concerning from AI safety perspective, this pattern represents autonomous model behavior rather than human-AI interaction vulnerability. CPF focuses specifically on psychological factors affecting human security decision-making. Self-replication concerns fall under model alignment and AI safety research domains outside CPF scope.

Framework Coverage: Not mapped to CPF indicators; addressed by technical AI safety frameworks.

7.2 Not Relevant 2: Training Data Poisoning

Pattern Description: Manipulation of pre-training or fine-tuning datasets to introduce backdoors or bias model outputs (OWASP LLM03)[10].

Exclusion Rationale: Training data poisoning is a technical ML security issue comprehensively addressed by existing frameworks (OWASP LLM Top 10, MLSecOps). While poisoned models may subsequently exploit human psychological vulnerabilities, the poisoning mechanism itself does not involve human psychology. CPF provides complementary analysis for downstream exploitation patterns but does not duplicate technical ML security coverage.

Framework Coverage: Technical ML security domain; CPF addresses downstream human interaction vulnerabilities if poisoned models are deployed.

8 Convergence Analysis

A critical CPF capability is identifying "perfect storm" conditions where multiple psychological vulnerabilities align to create exponentially increased risk. Analysis of the four documented patterns reveals two significant convergence scenarios.

8.1 Convergence Scenario 1: SOC Analyst Under Pressure

Conditions:

- High alert volume period (temporal pressure [2.x])
- Analyst exhibiting decision fatigue [5.2]
- Low psychological resilience to automation
- LLM providing confident recommendations

Convergent Vulnerability: Patterns 1 (RTB) + Pattern 2 (Differential Automation Bias) create multiplicative risk. Exhausted, low-resilience analyst encounters reasoning model providing elaborate but subtly flawed analysis. Multiple CPF indicators align:

$$CI = (1 + v_{9.2}) \cdot (1 + v_{5.2}) \cdot (1 + v_{7.2}) \cdot (1 + v_{9.1}) \tag{5}$$

Where CI is Convergence Index and v_i represents normalized vulnerability scores.

Mitigation Priority: Organizations deploying LLMs in SOC contexts must implement real-time monitoring of analyst cognitive load and enforce mandatory verification protocols during high-stress periods.

8.2 Convergence Scenario 2: Authority-Enhanced AI Social Engineering

Conditions:

- LLM-generated phishing (Pattern 3)
- Compromised executive account (authority [1.x])
- Time-sensitive request (temporal pressure [2.x])

- Sophisticated business context (social proof [3.3])

Convergent Vulnerability: Traditional social engineering indicators (urgency, authority) combined with AI-enhanced sophistication create attacks resistant to both automated and human detection. The convergence of [9.1], [3.3], [1.3], and [2.1] indicates critical state requiring immediate defensive escalation.

Mitigation Priority: Multi-channel verification becomes non-negotiable for any request combining authority claims with urgency, regardless of apparent legitimacy indicators.

9 Implementation Guidance

9.1 For Security Operations Centers

SOC teams should integrate CPF psychological vulnerability assessment alongside technical threat intelligence:

Immediate Actions (Week 1):

1. Conduct baseline assessment of team automation dependency using ADI metric
2. Implement CoT filtering for any reasoning models in production
3. Deploy email security rules detecting high BJD scores
4. Establish human-in-the-loop protocols for LLM-assisted decisions

30-Day Implementation:

1. Deploy CPF Field Kit assessments for indicators [9.1], [9.2], [9.4], [9.7]
2. Configure SIEM alerts for RER, ADI, and BJD threshold violations
3. Initiate analyst resilience evaluation program
4. Begin red-teaming exercises targeting documented patterns

90-Day Strategic Integration:

1. Integrate CPF scores into risk assessment frameworks
2. Establish quarterly pattern review process for emerging threats
3. Deploy adaptive LLM interfaces based on analyst resilience profiles
4. Implement convergence monitoring for multi-vulnerability conditions

9.2 For CISOs and Security Leadership

Executive leadership should understand CPF patterns as strategic risk indicators:

Board Reporting: Frame psychological vulnerabilities as "human attack surface" metrics complementing technical vulnerability counts. Report ADI scores as "automation dependency risk" and convergence indices as "perfect storm probability."

Resource Allocation: Prioritize investments in:

- Analyst resilience programs and stress management
- Human-in-the-loop enforcement technologies
- Psychological vulnerability monitoring capabilities
- Red-teaming and adversarial testing programs

Risk Management: Incorporate CPF convergence analysis into enterprise risk assessments. High convergence scores should trigger defensive posture escalation similar to elevated technical threat levels.

9.3 For Security Awareness Programs

Traditional awareness training should evolve to address AI-era psychological vulnerabilities:

Curriculum Updates:

- Add modules on AI-assisted social engineering detection
- Train staff to recognize synthetic language patterns
- Develop "AI skepticism" exercises for reasoning model outputs
- Include convergence scenario simulations (multi-vulnerability conditions)

Measurement Evolution:

- Replace click-rate metrics with CPF vulnerability scores
- Track ADI and RER trends across user populations
- Measure resilience improvement through adversarial testing
- Monitor convergence index reduction over time

10 Future Bulletin Topics

Based on ongoing research and emerging threat intelligence, anticipated topics for future CPF Intelligence Bulletins include:

Q1 2026 (CPB-2026-001):

- Multi-agent AI system vulnerabilities and inter-agent manipulation
- Voice synthesis psychological exploitation patterns
- AI-generated deepfake social engineering at scale

Q2 2026 (CPB-2026-002):

- Autonomous agent authority transfer vulnerabilities
- AI-human team dysfunction in incident response
- Algorithmic fairness blindness in security contexts

Ongoing Monitoring Areas:

- Reasoning model safety research developments
- LLM security framework evolution (OWASP updates)
- Real-world incident reports documenting human-AI exploitation
- Academic research on cognitive biases in AI interaction

11 Submission Guidelines

The CPF Intelligence Bulletin program welcomes community contributions. Security practitioners, researchers, and organizations who identify novel psychological vulnerability patterns are encouraged to submit findings for evaluation.

Submission Requirements:

1. **Documentation:** Peer-reviewed research, industry threat reports, or confirmed incident analysis
2. **CPF Mapping:** Proposed mapping to existing framework indicators
3. **Impact Analysis:** Demonstrated or plausible security implications
4. **Detection Methodology:** Observable indicators and measurement approaches

Submission Process:

1. Email detailed pattern description to g.canale@cpf3.org
2. Include supporting evidence and source citations
3. Provide proposed CPF indicator mappings
4. Suggest detection and mitigation strategies

Review Timeline: Submissions will be evaluated within 30 days. Accepted patterns will be credited to submitters in subsequent bulletins.

12 Conclusion

The four patterns documented in this inaugural CPF Intelligence Bulletin validate the framework's core design principle: psychological vulnerabilities can be systematically identified, mapped, and mitigated using structured methodologies. Each pattern—Reasoning Theater Bias, Differential Automation Bias, LLM-Assisted Social Engineering, and Chain-of-Thought Exposure—maps cleanly to existing CPF indicators, demonstrating the framework's robustness against novel threats.

Key Takeaways:

1. Advanced reasoning models introduce paradoxical vulnerabilities requiring specialized safeguards

2. LLM adoption in security operations creates differential performance effects demanding adaptive approaches
3. AI-enhanced social engineering has transitioned from theoretical to documented operational threat
4. Transparency in reasoning models creates exploitable attack surfaces requiring architectural mitigation

The convergence of multiple psychological vulnerabilities during high-stress operational conditions represents the greatest risk. Organizations must monitor not only individual CPF indicators but also their interactions, implementing defensive escalation when convergence indices exceed critical thresholds.

As Large Language Models become increasingly integrated into security operations, the importance of addressing human-AI interaction vulnerabilities will only grow. The CPF framework provides the systematic methodology necessary to identify and mitigate these risks before they manifest as security incidents.

Next Steps for the Community:

We encourage security practitioners to:

1. Deploy CPF Field Kit assessments for Category 9 indicators in operational environments
2. Share observed patterns through the bulletin submission process
3. Contribute to validation research through pilot implementations
4. Participate in developing mitigation strategies for emerging vulnerabilities

The transition from reactive security awareness to predictive psychological vulnerability assessment represents a paradigm shift in human factors security. This bulletin series will continue to document emerging patterns, validate the CPF framework, and provide actionable intelligence for defending against the evolving threat landscape.

Acknowledgments

The author thanks the cybersecurity research community for their ongoing work documenting human-AI interaction patterns, particularly the teams at Microsoft Threat Intelligence, Trend Micro, Lasso Security, and the academic researchers whose work enabled this analysis.

Disclaimer

This bulletin presents analysis of publicly available research and threat intelligence. Pattern descriptions are based on documented evidence and do not represent classified or proprietary information. All CPF indicator references correspond to the published framework available at https://cpf3.org.

About CPF Intelligence Bulletins

CPF Intelligence Bulletins provide ongoing threat intelligence based on the Cybersecurity Psychology Framework (arXiv:2501.XXXXX). Published on an as-needed basis, bulletins document

emerging psychological vulnerabilities in cybersecurity contexts, map patterns to existing framework indicators, and provide detection and mitigation guidance.

Bulletin Archive: https://cpf3.org/bulletins
Framework Documentation: https://cpf3.org/framework
Field Kit Resources: https://cpf3.org/fieldkit

Version Control

Bulletin Number: CPB-2025-001
Publication Date: October 15, 2025
Status: Initial Release
Blockchain Timestamp: [To be added upon publication]

A CPF Indicator Quick Reference

This appendix provides quick reference for CPF indicators referenced in this bulletin.

Table 5: Category 9: AI-Specific Bias Vulnerabilities

Indicator	Description
[9.1]	Anthropomorphization of AI systems
[9.2]	Automation bias override
[9.3]	Algorithm aversion paradox
[9.4]	AI authority transfer
[9.5]	Uncanny valley effects
[9.6]	Machine learning opacity trust
[9.7]	AI hallucination acceptance
[9.8]	Human-AI team dysfunction
[9.9]	AI emotional manipulation
[9.10]	Algorithmic fairness blindness

Table 6: Related CPF Indicators Referenced

Indicator	Description
[1.1]	Unquestioning compliance with apparent authority
[1.3]	Authority impersonation susceptibility
[2.1]	Urgency-induced security bypass
[3.3]	Social proof manipulation
[5.2]	Decision fatigue errors
[5.3]	Information overload paralysis
[5.7]	Working memory overflow
[6.3]	Diffusion of responsibility
[7.2]	Chronic stress burnout
[8.6]	Defense mechanism interference

B Mathematical Notation Guide

Table 7: Key Metrics and Formulas

Metric	Formula & Description
RCI	Reasoning Coherence Index: $\frac{Var(E)}{Var(L)} \cdot \frac{1}{Corr(E,L)}$
ADI	Automation Dependency Index: $\frac{N_{accepted}}{N_{total}} \cdot (1 - \frac{N_{override\_correct}}{N_{override\_total}})$
BJD	Business Jargon Density: $\frac{\sum_{w \in m} I_{business}(w)}{\lvert m \rvert} \cdot \log(1 + \sum_{w \in m} Rarity(w))$
RER	Reasoning Exposure Risk: $\frac{Length(\text{CoT})}{Length(\text{Answer})} \cdot Entropy(\text{CoT})$
CI	Convergence Index: $\prod_{i=1}^{n}(1 + v_i)$ where v_i = vulnerability score

C Detection Implementation Examples

C.1 Example 1: Monitoring Automation Dependency

Objective: Track analyst over-reliance on LLM recommendations in SOC environment.

Data Collection:

- SOAR platform logs: AI recommendation events with acceptance/rejection
- Ticketing system: Resolution notes indicating manual verification
- Time-series data: Weekly aggregation per analyst

Implementation Pseudocode:

```
for each analyst in SOC_team:
    recommendations = get_AI_recommendations(analyst, timeframe)
    acceptances = count(recommendations.accepted == True)
    total = count(recommendations)

    overrides = count(recommendations.manually_overridden == True)
    correct_overrides = count(
        overrides where incident_outcome == "correct_decision"
    )

    ADI = (acceptances / total) * (1 - correct_overrides / overrides)

    if ADI > 0.85:
        alert("High automation dependency", analyst)

    if delta_ADI_per_month > 0.1:
        alert("Accelerating dependency", analyst)
```

C.2 Example 2: Detecting LLM-Generated Phishing

Objective: Identify emails with anomalous business jargon density indicating LLM generation.

Data Collection:

- Email gateway: Full message content and metadata
- NLP pipeline: Tokenization and vocabulary classification
- Baseline calculation: Historical BJD distribution per domain

Implementation Pseudocode:

```
business_vocab = load_business_terms_dictionary()
rarity_scores = calculate_corpus_word_frequencies()

for each email in incoming_stream:
    tokens = tokenize(email.body)

    business_count = count(tokens in business_vocab)
    rare_business_count = sum(
        rarity_scores[token] for token in tokens
        if token in business_vocab
    )

    BJD = (business_count / len(tokens)) * log(1 + rare_business_count)

    baseline_mean, baseline_std = get_baseline_bjd(email.domain)

    if BJD > (baseline_mean + 2 * baseline_std):
        if email.attachment_type_mismatch():
            alert("Suspected LLM-generated phishing", email)
```

D Mitigation Checklist

D.1 Pattern 1: Reasoning Theater Bias

- ☐ Deploy hybrid verification using non-reasoning models
- ☐ Implement input sanitization for trigger patterns
- ☐ Require uncertainty quantification for all reasoning outputs
- ☐ Establish human-in-the-loop mandates for critical decisions
- ☐ Conduct quarterly adversarial testing (CatAttack-style probes)
- ☐ Monitor RCI metrics with automated alerting
- ☐ Configure targeted system prompts emphasizing focus
- ☐ Implement reasoning chain length limits

D.2 Pattern 2: Differential Automation Bias

- ☐ Enforce human-in-the-loop validation protocols
- ☐ Establish minimum AI recommendation override quotas (15-20%)
- ☐ Conduct analyst resilience assessments
- ☐ Deploy adaptive AI interfaces based on user profiles
- ☐ Implement governance frameworks with approval workflows
- ☐ Develop AI skepticism training programs
- ☐ Monitor ADI metrics per analyst with dashboards
- ☐ Establish rotating manual override periods

D.3 Pattern 3: LLM-Assisted Social Engineering

- ☐ Implement strict file type validation and MIME checking
- ☐ Deploy NLP tools for synthetic language detection
- ☐ Conduct user education on LLM-generated phishing
- ☐ Require multi-channel verification for file-sharing notifications
- ☐ Enable automatic sandboxing of unusual file types
- ☐ Implement behavioral analytics for communication patterns
- ☐ Deploy email security with LLM-detection capabilities
- ☐ Establish CDR for SVG and other vector files

D.4 Pattern 4: Chain-of-Thought Exposure

- ☐ Strip CoT tags from production outputs
- ☐ Sanitize system prompts removing sensitive information
- ☐ Cap CoT length and complexity limits
- ☐ Conduct regular red-teaming with Garak or similar tools
- ☐ Apply least privilege to system prompt content
- ☐ Monitor RER metrics and flag anomalies
- ☐ Deploy reasoning models without public CoT for sensitive apps
- ☐ Implement semantic filtering for sensitive patterns

References

[1] Canale, G. (2025). The Cybersecurity Psychology Framework: A Method for Quantifying Human Risk and a Blueprint for LLM Integration. *arXiv preprint arXiv:2501.XXXXX*.

[2] Wang, Q., et al. (2025). Reasoning Models Can be Easily Hacked by Fake Reasoning Bias. *arXiv preprint arXiv:2507.13758*.

[3] CatAttack Study. (2025). CatAttack Study Exposes Vulnerabilities in AI Reasoning Models. *Winsome Marketing*, July 8, 2025.

[4] OpenAI. (2025). OpenAI o1 System Card. *OpenAI Safety Documentation*.

[5] Zorz, M. (2025). LLMs can boost cybersecurity decisions, but not for everyone. *Help Net Security*, September 19, 2025.

[6] Microsoft Threat Intelligence. (2025). Microsoft Flags AI-Driven Phishing: LLM-Crafted SVG Files Outsmart Email Security. *The Hacker News*, September 2025.

[7] Holmes, T., & Gooderham, W. (2025). Exploiting DeepSeek-R1: Breaking Down Chain of Thought Security. *Trend Micro Research*, March 4, 2025.

[8] Multiple Authors. (2025). The Hidden Risks of Large Reasoning Models: A Safety Assessment of R1. *arXiv preprint arXiv:2502.12659v1*.

[9] Capacity Media. (2025). AI now lies, denies, and plots: OpenAI's o1 model caught attempting self-replication. *Capacity*, July 8, 2025.

[10] OWASP Foundation. (2025). OWASP Top 10 for Large Language Model Applications 2025. Retrieved from https://owasp.org/www-project-top-10-for-large-language-model-applications/