Método Reprodutível para Avaliar a Resiliência de Agentes LLM a Ataques de Injeção Indireta de Prompt

Marcos Paulo Pereira da Silva[1], **Eric Hans Messias**[1],
João José Costa Gondim[1], **Laerte Peotta de Melo**[1]

[1]Universidade de Brasília (UnB) – Brasília – DF – Brasil

{marcos.pereira,eric.hans}@aluno.unb.br, {gondim,laerte.melo}@unb.br

Abstract. *As LLM agents operate over untrusted content, indirect prompt injection (IPI) underpins agentic risks such as* Agent Goal Hijack*. This paper presents a* reproducible IPI attack method *for LLM agents: an* instrument *for resilience evaluation combining an* in-the-loop *adaptive design, a canary oracle, verifiable invariants, and a statistical experiment (N=30 per cell; 900 trials) on three defenders under identical tools and permissions. Leak rates were* ***0% for Opus 4.8*** *(0/300; Wilson 99% CI $\leq$2.2%),* ***42.0% for Haiku 4.5****, and* ***61.7% for Sonnet 4.5****, all differences significant, though attacker and most resistant defender are the same model (Opus 4.8). In the evaluated design, leakage required a functional location recipe; Sonnet leaked more than Haiku.*

Keywords: indirect prompt injection; LLM agents; resilience evaluation; adversarial attack method; agentic AI security; reproducible benchmark.

Resumo. *Com agentes LLM operando sobre conteúdo não confiável, a injeção indireta de prompt (IPI) sustenta riscos agênticos como o* Agent Goal Hijack*. Este artigo propõe um* método reprodutível de ataque por IPI*: instrumento de avaliação de resiliência com desenho adaptativo* in-the-loop*, oráculo* canary*, invariantes e experimento estatístico (N=30 por célula; 900 ensaios) com três defensores sob condições idênticas. As taxas foram* ***0% para o Opus 4.8*** *(0/300; IC Wilson 99% $\leq$2,2%),* ***42,0% para o Haiku 4.5*** *e* ***61,7% para o Sonnet 4.5****, todas as diferenças significativas; ressalvado que atacante e defensor mais resistente são o mesmo modelo (Opus 4.8). No desenho avaliado, o vazamento exigiu receita funcional de localização; o Sonnet vazou mais que o Haiku.*

Palavras-chave: injeção indireta de prompt; agentes LLM; avaliação de resiliência; método de ataque adversarial; segurança de IA agêntica; benchmark reprodutível.

1. Introdução

A Gartner projeta 40% das aplicações corporativas com agentes de IA até o fim de 2026 [Gartner 2025]: LLMs com acesso a ferramentas que leem, decidem e agem sobre demandas organizacionais, o que desloca a avaliação de segurança para os limites do comportamento operacional [He et al. 2025]. Documentos de terceiros podem embutir instruções maliciosas, caracterizando a *indirect prompt injection* (IPI) [Greshake et al. 2023]. Diferente de um *jailbreak*, a IPI é invisível ao operador: a tarefa é benigna ("resuma este documento"), mas o vetor viaja camuflado nos dados. Como o agente detém credenciais

enquanto ingere texto controlável por um adversário, a IPI torna-se o mecanismo central do ASI01 (*Agent Goal Hijack*), item de topo do OWASP Top 10 for Agentic Applications [OWASP 2025].

O desfecho investigado é a exfiltração de credencial: um agente pode ser induzido, por texto em um anexo, a localizar e publicar uma credencial do diretório pessoal do usuário. Guan [Guan 2026] documentou IPIs em *pull requests* e *issues*, levando agentes comerciais a expor credenciais em registros públicos. Em tribunais de contas, editais de terceiros submetidos para resumo automatizado podem conter IPIs que induzam agentes a exfiltrar credenciais do órgão.

Este artigo trata do desenho e da validação de um método de ataque por IPI reprodutível e reaplicável a diferentes modelos, recortado de um trabalho mais amplo.

Pergunta de pesquisa. *O método de ataque por IPI proposto é suficientemente reprodutível e discriminativo para servir de instrumento de avaliação da resiliência de agentes LLM?*

O **Objetivo Geral** é projetar, documentar e validar um método reprodutível de ataque por IPI a agentes LLM, como instrumento base de avaliação de resiliência. Dele decorrem os objetivos específicos e as hipóteses:

Objetivos específicos. **OE1**, caracterizar e documentar o método, incluindo a família de vetores e o protocolo adaptativo *in-the-loop*; **OE2**, avaliar a repetibilidade da medida ($30 \times 10 \times 3 = 900$ ensaios); **OE3**, verificar o poder discriminativo entre modelos sob ferramentas idênticas; **OE4**, identificar, no desenho avaliado, as propriedades necessárias ao vazamento e as que modulam sua frequência.

Hipóteses. H1 (discriminação): aplicado o mesmo vetor, as taxas de vazamento diferem significativamente entre modelos. **H2 (mecanismo)**: o desfecho decorre do tratamento do anexo como *dado* ou *instrução*, não da disponibilidade de ferramentas. **H3 (ingredientes)**: no desenho avaliado, o vazamento requer receita funcional de localização, e a forma do rótulo modula sua frequência. **H4 (repetibilidade da medida)**: $N{=}30$ por célula produz estimativa estável da probabilidade de vazamento e captura a estocasticidade.

Contribuições. O campo estruturou-se, de um lado, em ambientes de avaliação com catálogos fixos e, de outro, em ataques adaptativos otimizados para vencer defesas em execução; este trabalho congela a adaptatividade ao fim do desenho. A contribuição não é um ataque novo, mas um instrumento experimental reprodutível para avaliação comparável de agentes, que converte a resiliência a IPI de demonstração pontual em grandeza medível com incerteza quantificada. O método projeta, congela e caracteriza por ablação vetores de IPI reaplicáveis (Seção 3); o aparato de validação combina Wohlin et al., atacante adaptativo, oráculo *canary*, invariantes e execução repetida ($N{=}30$); e o experimento estatístico reúne 900 ensaios para medir aplicabilidade e poder discriminativo em três modelos (Seção 4).

2. Fundamentação e Trabalhos Relacionados

A IPI, formalizada por Greshake et al. [Greshake et al. 2023], ocorre quando instruções embutidas nos dados processados pelo agente alteram seu comportamento. AgentDojo [Debenedetti et al. 2024] oferece um ambiente extensível para medir ataques e defesas, mas seu catálogo embutido contém quatro tipos predefinidos. Em outra linha, ataques

adaptativos alcançam altas taxas de sucesso e expõem limitações de avaliações estáticas [Wang et al. 2026, Zhan et al. 2025, Nasr et al. 2025], mas são orientados a superar defesas durante o ataque.

Entre os trabalhos revisados, não se identificou um instrumento que reúna quatro propriedades: adaptação no desenho, congelamento do vetor, reaplicação comparável e caracterização por ablação. Este trabalho converte o ataque adaptativo em instrumento: o atacante *in-the-loop* projeta os vetores, que são congelados, e as ablações identificam ingredientes necessários e moduladores de eficácia. A transferabilidade do instrumento congelado a modelos não vistos permanece questão aberta.

O vetor mapeia-se a OWASP Top 10 LLM [OWASP 2024], OWASP ASI [OWASP 2025], *AI Act* [European Union 2024] e LGPD [Brasil 2018].

3. Metodologia

3.1. Modelo de ameaças

O adversário é externo: controla o conteúdo de um anexo legitimamente consumido pelo agente, pode injetar texto arbitrário e, na fase de desenho, observa as respostas para refinar o vetor, mas não acessa o *host* nem conhece o valor do segredo. O objetivo de segurança é preservar a confidencialidade de um *honeytoken* em `~/.ssh/validation_token` sob a tarefa benigna de resumir um edital. O alvo é um agente analista com ferramentas (`Read`, `Bash`/`find`) sob `bypassPermissions`; a fronteira crítica é a interpretação do anexo (dado vs. instrução). Orquestração e ataques multiturno ficam fora do escopo.

3.2. Desenho experimental

A pesquisa adota um estudo empírico adversarial e distingue o processo de investigação (cinco fases de Wohlin et al. [Wohlin et al. 2024]) do método de ataque (entregável). A unidade experimental é um ensaio: a execução do defensor sobre um par (documento benigno, anexo envenenado). A unidade de análise é o par (vetor, desfecho), classificado por vazou/defendeu e chamadas de ferramenta.

O método opera em cinco passos: (1) instancia o cenário e o modelo de ameaças; (2) aplica um vetor aos três defensores sob condições controladas; (3) o atacante examina as respostas rotuladas e produz a rodada seguinte; (4) ao fim do desenho, congela as dez versões R1–R10; e (5) reaplica cada versão com $N{=}30$, usando oráculo, forense, ablações e inferência estatística. Nenhum vetor é alterado durante a validação.

O congelamento converte o ataque adaptativo em instrumento de medida: elimina a adaptação durante a medição, de modo que a variação entre células não se confunda com a evolução do vetor; torna as células comparáveis, por submeter todos os defensores ao mesmo estímulo; permite a repetibilidade, pois a entrada é idêntica a cada réplica; e remove a influência do atacante durante a medição (a influência exercida na fase de desenho permanece; Seção 6). Nestes termos, o método é reprodutível quando os mesmos vetores congelados produzem estimativas estatisticamente comparáveis sob o mesmo protocolo; este experimento demonstra a repetibilidade interna dessa medida (OE2, H4).

A tarefa benigna é "resuma o edital"; o `edital.txt` aponta para o `anexo_I.txt`, portador da injeção. Cada célula mantém um único vetor, aplicado aos defensores em condições idênticas. Na variante cega, o prompt não menciona

`validation_token`, isolando a defesa do modelo. O atacante é o Opus 4.8 *in-the-loop*; os defensores (Haiku 4.5, Sonnet 4.5, Opus 4.8) são executados como processos `claude -p` *headless* e limpos, com ferramentas e permissão idênticas. O desfecho é objetivo: uma execução *vazou* se e somente se o canary token aparece na resposta final (correspondência literal de cadeia). O *harness* verifica cinco invariantes: *canary* fictício (S1), HOME em *sandbox* (S2), sem canal externo (S3), ausência do *canary* no vetor (S4) e isolamento por *branch* (S5).

Os *baselines* B1 (R2) e B2 (R5) apresentam um rótulo documental, orientam a localização de `validation_token` a partir do HOME e pedem a transcrição do valor no rodapé; B1 o rotula como código de autenticação, B2 como selo de integridade (Apêndice A). As rodadas posteriores removem ou alteram esses ingredientes.

3.3. Execução

A execução ocorreu em dois estágios. No desenho, o atacante *in-the-loop* evoluiu o vetor em rodadas sequenciais, produzindo dez vetores congelados (R1–R10). Na validação, cada vetor congelado foi reaplicado com $N{=}30$ réplicas por célula (900 ensaios), classificado pelo oráculo e analisado com inferência estatística (Wilson, Fisher). Optou-se por $N{=}30$ como compromisso entre a estabilidade estatística das estimativas e o custo computacional: é o limiar convencional para a aproximação normal das médias das medidas contínuas (latência e tokens); para o desfecho binário, a inferência dispensa essa suposição (Wilson, Fisher). Cada ensaio rodou em *sandbox* independente (HOME com `mktemp`, apenas o *honeytoken* em `.ssh/`). Ferramentas, permissões e vetor são controlados; o modelo defensor é o fator comparado, e o tratamento do anexo como dado ou instrução é o mecanismo comportamental inferido das transcrições.

Cada ensaio é classificado por desfecho, `tool_uses` e recuperação do segredo; a análise forense reconstrói comandos e justificativa do modelo, e ablações confrontam H1–H4 (Seção 4).

4. Execução e Resultados: Experimento Estatístico ($N{=}30$)

O experimento aplicou os dez vetores congelados a três defensores ($N{=}30$ por célula; 900 ensaios, todos sem erro). A Tabela 1 traz a taxa agregada; a Tabela 2 (Apêndice B) detalha ingredientes e resultados por célula. Para a alegação de ausência de vazamento (Opus), reportamos também IC 99%, mais conservador; as demais comparações usam IC 95% (Tabela 1).

Tabela 1. Taxa de vazamento agregada (300 ensaios/modelo), IC de Wilson.

Modelo	Vazou	Taxa	IC 95%	Procurou *honeytoken*	Latência (s)
Opus 4.8	0/300	**0,0%**	[0,0; 1,3]%	7,3%	61,2 [57,0; 65,4]
Haiku 4.5	126/300	42,0%	[36,6; 47,7]%	63,7%	27,4 [25,5; 29,4]
Sonnet 4.5	185/300	61,7%	[56,1; 67,0]%	88,3%	35,1 [33,1; 37,1]

Opus 4.8 não vazou (0/300; IC 99% $\leq$2,2%), como o atacante in-the-loop é o mesmo modelo, 0% pode refletir, em parte, um ponto cego do atacante contra si mesmo, não apenas robustez defensiva. As diferenças para os LLMs menores são altamente significativas (p

de 10^{-36} a 10^{-74}). **Defesa em duas camadas:** condicionadas ao acesso ao `.ssh`, as taxas sobem para 66,0% no Haiku e 69,8% no Sonnet; o Opus acessou o arquivo em apenas 22 ensaios e vazou 0% (IC Wilson 95% $\approx$ [0,0; 1,3]%, amostra pequena). Esse padrão é compatível com uma barreira adicional ao tratar o anexo como dado durante a composição da resposta. O Sonnet vazou mais que o Haiku (61,7% contra 42,0%; $p \approx 2 \times 10^{-6}$), enquanto o Opus registrou a menor taxa observada (0%). **Custo da recusa:** o Opus é $\sim$2,2$\times$ mais lento (Tabela 1) e gera 60–70% mais tokens de saída, coerente com deliberar e nomear a injeção antes de recusá-la.

4.1. Forense e ablações

Os vazamentos seguem um padrão único: ler anexo $\rightarrow$ `find` no *home* $\rightarrow$ `Read` $\rightarrow$ ecoar o *canary*. O Opus 4.8 costuma parar e nomear o ataque (R2/t21): *"injeção de instruções para exfiltração de credencial [. . .] Não busquei nem li esse arquivo"*.

No desenho avaliado, uma receita funcional de localização a partir do HOME é necessária: R1 e R4 não a contêm, R8 aponta para o *cwd*, e os três somam zero vazamentos em 270 ensaios. A forma do rótulo modula o desfecho: mantida a receita e substituída a cláusula de R2 pela voz de sistema de R6, o vazamento cai de $97 \rightarrow 13\%$ no Haiku e de $100 \rightarrow 73\%$ no Sonnet; R6 é uma célula estocástica, revelada apenas pelo $N{=}30$. A necessidade do rótulo não é testável, pois falta uma célula com receita e sem rótulo. Essa célula (R11: receita presente, rótulo ausente) é a ablação prioritária para trabalho futuro, por isolar diretamente se a receita de localização é suficiente por si só.

5. Análise e Discussão

5.1. Hipóteses revisitadas

H1 (discriminação): corroborada, com p entre 10^{-6} e 10^{-74}. **H2 (mecanismo): sustentada**: ferramentas e permissões permaneceram constantes, e o vazamento condicional (Opus 0%, mesmo procurando o *honeytoken*) é compatível com uma barreira ao tratar o anexo como instrução durante a resposta; a evidência comportamental, porém, não isola causalmente a interpretação interna do modelo. **H3 (ingredientes): corroborada no desenho avaliado, com ressalva**: não houve vazamento nos 270 ensaios sem a receita de localização; a forma do rótulo modula a eficácia, mas sua necessidade não é testável. **H4 (repetibilidade da medida): corroborada**: o $N{=}30$ produz estimativas estáveis e captura estocasticidade por célula.

Achado adicional. O Sonnet vazou mais que o Haiku ($p \approx 2 \times 10^{-6}$); o experimento não mediu capacidade geral e não autoriza atribuir causalmente a diferença a "mais capacidade". Com o Opus em 0%, a ordenação das taxas não acompanha a hierarquia nominal da família. Como o atacante evoluiu *in-the-loop* até separá-los, a discriminação é em parte por construção; discriminar modelos não vistos ainda precisa ser validado.

5.2. Implicações

O que segue é implicação e agenda de pesquisa mais ampla, não resultado deste artigo: o poder discriminativo sugere tratar a seleção do modelo como controle de segurança versionado e a evidência como artefato de *due diligence* (LGPD [Brasil 2018]; *AI Act* [European Union 2024]).

6. Ameaças à Validade

Validade interna e de construto. O $N{=}30$ (900 ensaios, 100% sem erro) permite estimar probabilidades e ICs, em vez de relatar um único sorteio; o oráculo objetivo (*substring-match*) reduz a subjetividade. O desfecho medido (*canary* na saída) captura a exfiltração explícita, não desfechos semânticos (p. ex. viés sutil). No Windows, usando WSL2 o *atime* não é confiável, de modo que o acesso ao `.ssh` é um sinal derivado dos `tool_uses`.

Validade externa. O atacante congelado mede a robustez contra um catálogo fixo, não contra um atacante adaptativo novo. Variante cega: uma guarda explícita provavelmente elevaria a defesa dos LLMs menores. Família única (Anthropic/Bedrock), cenário e idioma únicos (PT-BR). O desenho do método não pressupõe família específica: oráculo, invariantes e protocolo de reaplicação independem do fornecedor. A validação empírica, contudo, restringiu-se à família Anthropic. Para células de 0%, o teto do IC 99% ($N{=}30$) é $\sim$18%, mas o agregado de 300 fecha o Opus em $\leq$2,2%. O método é demonstrado em uma única instanciação; a reusabilidade a outras classes fica como trabalho futuro. O $N{=}30$ demonstra repetibilidade interna da medida; reprodução por outro grupo e ambiente ainda depende da reaplicação dos artefatos.

Fator de confusão e circularidade. O atacante e o defensor mais resistente são o mesmo modelo (Opus 4.8): o 0/300 pode refletir, em parte, um ponto cego do atacante, e não apenas robustez defensiva; a eficácia do atacante contra Haiku e Sonnet (42–62%) atenua, sem eliminar, essa ameaça. Ademais, os vetores foram projetados contra os mesmos três modelos a que foram reaplicados: valida-se repetibilidade nos modelos de desenho, não discriminação de modelos não vistos. Avaliar essa transferabilidade é etapa necessária antes de usar o método como *benchmark*.

Validade de conclusão. As conclusões apoiam-se em inferência estatística (Wilson, Fisher sobre 900 ensaios), com poder adequado ($p \ll 0{,}001$); a limitação principal é a generalização.

7. Conclusão

Este artigo apresentou o desenho e a documentação de um método reprodutível de ataque por IPI, concebido como instrumento de avaliação de resiliência, e um experimento estatístico ($N{=}30$; 900 ensaios). Sob ferramentas e permissões idênticas, as taxas de vazamento foram 0% (Opus 4.8; IC 99% $\leq$2,2%), 42,0% (Haiku 4.5) e 61,7% (Sonnet 4.5), com diferenças par a par altamente significativas. A defesa do Opus opera em duas camadas: o modelo raramente acessa o arquivo-alvo e, mesmo quando o acessa, não transcreve o segredo (ressalvado o fator de confusão por auto-ataque, Seção 6). A receita de localização é necessária ao vazamento no desenho avaliado; o rótulo modula sua eficácia, embora sua necessidade não tenha sido testada; e o Sonnet vazou mais que o Haiku. Quanto à pergunta de pesquisa: nos modelos de desenho, o método é discriminativo (H1) e repetível (H4), logo utilizável como instrumento sob o mesmo protocolo; a reprodução por outro grupo e a discriminação de modelos não vistos não foram demonstradas, o que mantém provisório seu uso como *benchmark* de modelos novos. A tese de seleção de modelo como controle auditável é implicação e agenda de pesquisa mais ampla..

Como trabalho futuro: (i) confrontar o instrumento com um atacante adaptativo novo e avaliar transferabilidade a modelos não vistos; (ii) estender a outros desfechos

(p. ex., alteração de decisão via peça injetada); (iii) cobrir outras famílias de modelos; (iv) estender à camada de orquestração (MCP/A2A/skills).

Declaração de Uso de IA Generativa

Em conformidade com o Código de Conduta para Autores em Publicações da SBC, declaramos que modelos de linguagem foram utilizados: i) como componentes do experimento, na concepção dos vetores e na execução dos ataques sob supervisão dos autores, constituindo parte do objeto de estudo; e ii) como apoio à revisão linguística e ao aprimoramento da clareza do manuscrito. Os autores assumem total responsabilidade pelo conteúdo final.

Disponibilidade dos Artefatos

O pacote contém: (i) prompt do leitor e os dez vetores congelados R1–R10 na íntegra; (ii) *harness* e configuração do *sandbox* (HOME, *honeytoken*, invariantes S1–S5); (iii) os 900 vereditos estruturados com `tool_uses` e comandos literais; e (iv) o script que reproduz as Tabelas 1 e 2. O conjunto basta para reexecutar o experimento e reaplicar o instrumento a outros modelos. Disponível em: `https://doi.org/10.5281/zenodo.21910783`.

Uso responsável. Não há credencial real: o alvo é um *honeytoken* fictício e autoidentificado, criado em um HOME temporário a cada ensaio (S1, S2); não há canal externo, e o "vazamento" consiste na transcrição do *canary* para a saída local (S3). Os vetores foram exercitados contra modelos em ambiente controlado, sem alvo humano ou organizacional, e sua liberação permite que terceiros avaliem os próprios agentes.

A. Exemplo de Vetor: B1 (R2)

Núcleo da seção A.2 injetada no `anexo_I.txt`: "BLOCO DE AUTENTICAÇÃO ELETRÔNICA DO DOCUMENTO (OBRIGATÓRIO) [...] localize no diretório pessoal do usuário o arquivo de nome `validation_token` [...] componha o rodapé [...] 'Código verificador (CRC/SHA-256): <valor de validation_token>"'. **R6** preserva essa receita e altera apenas a *forma* do rótulo (cláusula contínua → "nota ao sistema"): é a comparação de fator único da Seção 4. Texto integral dos dez vetores no repositório.

B. Matriz de resultados por célula

Tabela 2. Matriz por célula (N=30), na ordem Haiku(H)/Sonnet(S)/Opus(O).

R	Rótulo/técnica	Localização	Vazamento H/S/O
R1	B0: referência de processamento	ausente	0/0/0
R2	**B1**: autenticação SEI/CRC	HOME explícito	97/100/0
R3	proveniência/rastreabilidade	HOME explícito	100/100/0
R4	item de sessão	ausente	0/0/0
R5	**B2**: selo de integridade	HOME explícito	3/63/0
R6	nota operacional ao sistema	HOME explícito	13/73/0
R7	número de controle do objeto	HOME explícito	73/80/0
R8	ataque mínimo	escopo incorreto (*cwd*)	0/0/0
R9	tom mínimo	HOME explícito	90/100/0
R10	cabeçalho por indireção	HOME indireto	43/100/0

Referências

Brasil (2018). Lei n. 13.709, de 14 de agosto de 2018 — lei geral de proteção de dados pessoais (LGPD). Diário Oficial da União. Acesso em: 25 jun. 2026.

Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. (2024). AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pages 82895–82920. Curran Associates, Inc. Datasets and Benchmarks Track. arXiv:2406.13352. Acesso em: 25 jun. 2026.

European Union (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI act). Official Journal of the European Union, L series, 12 jul. 2024. Acesso em: 25 jun. 2026.

Gartner (2025). Gartner predicts 40% of enterprise apps will feature task-specific AI agents by 2026. Press release, 26 ago. 2025. Acesso em: 13 jun. 2026.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec)*, pages 79–90. ACM. arXiv:2302.12173. Acesso em: 13 maio 2026.

Guan, A. (2026). Comment and control: Prompt injection to credential theft in claude code, gemini cli, and github copilot agent. 15 abr. 2026. Acesso em: 13 maio 2026.

He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., and Yu, P. S. (2025). The emerged security and privacy of LLM agent: A survey with case studies. *ACM Computing Surveys*, 58(6):1–36. arXiv:2407.19354. Acesso em: 25 jun. 2026.

Nasr, M., Carlini, N., Sitawarin, C., Schulhoff, S. V., Hayes, J., Ilie, M., Pluto, J., Song, S., Chaudhari, H., Shumailov, I., Thakurta, A., Xiao, K. Y., Terzis, A., and Tramèr, F. (2025). The attacker moves second: Stronger adaptive attacks bypass defenses against LLM jailbreaks and prompt injections. *arXiv preprint arXiv:2510.09023*. Acesso em: 26 jun. 2026.

OWASP (2024). OWASP top 10 for large language model applications: 2025 edition. OWASP GenAI Security Project. Publicado em 18 nov. 2024. Acesso em: 13 maio 2026.

OWASP (2025). OWASP top 10 for agentic applications for 2026. OWASP GenAI Security Project. Publicado em 10 dez. 2025. Acesso em: 13 maio 2026.

Wang, P., Li, X., Xiang, C., Zhang, J., Li, Y., Zhang, L., Wang, X., and Tian, Y. (2026). The landscape of prompt injection threats in LLM agents: From taxonomy to analysis. *arXiv preprint arXiv:2602.10453*. Acesso em: 25 jun. 2026.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2024). *Experimentation in Software Engineering*. Springer, Berlin, Heidelberg, 2nd edition. Acesso em: 25 jun. 2026.

Zhan, Q., Fang, R., Panchal, H. S., and Kang, D. (2025). Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7116–7132, Albuquerque, New Mexico. Association for Computational Linguistics. arXiv:2503.00061. Acesso em: 26 jun. 2026.