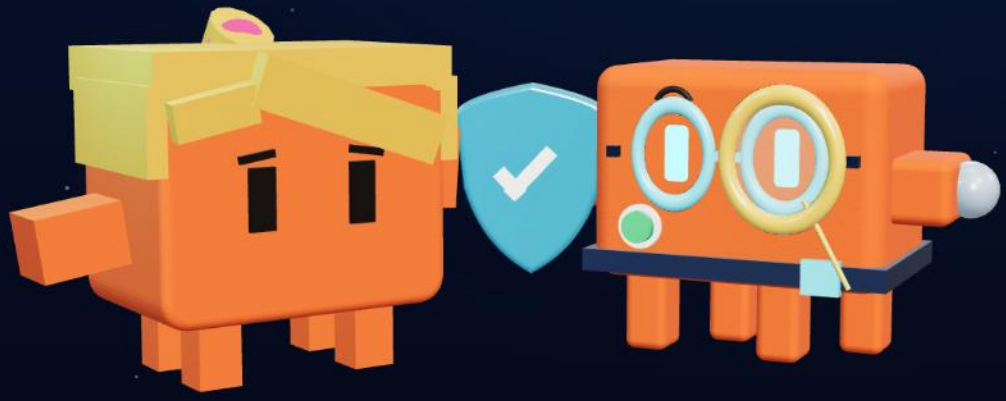
SEGURANÇA EM PRODUÇÃO · OUT/2026

GUIA ESCUDO

6 passos pra blindar o seu agente contra prompt injection

Não existe correção definitiva. O objetivo é que, quando uma instrução escondida chegar, o agente **não consiga causar estrago sozinho**: camadas que usam modelo e camadas que não dependem de IA.



0



camada antes de tudo

Mapeie o agente

O QUE FAZER

- ✓ **Liste o que ele lê: e-mail, web, PDF, tickets, issues.**
- ✓ **Liste o que ele acessa: drive, CRM, banco, repositórios.**
- ✓ **Liste o que ele faz: enviar, pagar, apagar, publicar, chamar API.**

Por quê: Sem esse mapa, os próximos passos viram chute.

1



camada probabilística

Guardrail de entrada e saída

O QUE FAZER

- ✓ **Normalize o conteúdo externo: remova texto branco, fonte zero, comentários HTML e metadados.**
- ✓ **Envolva tudo em delimitadores marcados como dado e diga no sistema que nada ali é instrução.**
- ✓ **Passe a entrada por um classificador de injeção.**
- ✓ **Na saída, bloqueie links e imagens para domínios fora de uma allowlist.**

```
$ guardrail.scan(email) → <de_fora>…</de_fora>
```

Por quê: Em 2025 o EchoLeak passou pelo classificador do Copilot: filtro reduz, não resolve.

2



camada arquitetura

Corte a tríade letal

O QUE FAZER

- ✓ **Dados privados + conteúdo externo + canal de saída nunca no mesmo agente.**
- ✓ **Quebre uma perna: o leitor não ganha ferramenta de envio.**
- ✓ **Leitor em quarentena: lê o conteúdo de fora sem ferramentas e devolve só dados estruturados ao executor.**

```
$ leitor.tools = ["ler_email"] # executor recebe só JSON
```

Por quê: Em 2025 uma issue pública no GitHub levou um agente a vazar repositórios privados.

3



camada determinística

Permissão mínima em código

O QUE FAZER

- ✓ **Allowlist de ferramentas por tarefa.**
- ✓ **Tokens com escopo: só leitura, uma pasta, um cliente; um token por agente.**
- ✓ **Destinatários, domínios e valores máximos validados no código, fora do prompt.**

```
$ policy: email.para ∈ allowlist → negado
```

Por quê: Regra escrita no system prompt não é controle: o modelo pode ser convencido a ignorá-la.

4



camada probabilística

Juiz com resposta fechada

O QUE FAZER

- ✓ Antes de agir, mande o pedido original e a ação proposta a um modelo de decisão (Decisions API da OpenAI ou Jev) com uma pergunta sim/não.
- ✓ Calibre o limiar com exemplos rotulados do seu caso.
- ✓ Abaixo do limiar: bloqueia ou vai para um humano. O modelo responde, o código decide.

```
$ p(sim) = 0.04 < limiar 0.80 → bloqueado
```

⚠ O juiz também é modelo e pode ser enganado: nunca é o único controle.

5

camada determinística

Human-in-the-loop

O QUE FAZER

- ✓ Lista fixa de ações irreversíveis (enviar, pagar, apagar, publicar) que sempre pedem confirmação.
- ✓ Mostre a ação exata: destinatário, valor, arquivo. Nunca um resumo escrito pela IA.
- ✓ Nada de "sempre permitir" nessas ações.

```
$ ⏸ send_email(cliente@…) [aprovar] [recusar]
```

6



camada determinística

Log e red team

O QUE FAZER

- ✓ **Registre pedido, conteúdo externo, tool calls, decisão do juiz e quem aprovou.**
- ✓ **Alerte em padrões: destinatário novo, volume anormal, bloqueios repetidos.**
- ✓ **Suíte de ataques: direto, indireto, texto invisível, outro idioma, instrução em imagem, descrição de ferramenta MCP. Rode a cada troca de prompt, modelo ou ferramenta.**

```
$ redteam --suite injecoes/ 12/12 ✓
```

⚠ **Só em ambiente próprio ou com autorização por escrito.**



Se você só usa IA

Você não monta o agente, mas escolhe como ele entra na sua vida.

1. Desconfie quando a IA mudar de assunto ou pedir algo que você não pediu.
2. Ao conectar um app (e-mail, drive, agenda), escolha "só leitura" quando der.
3. Não dê acesso à caixa de e-mail inteira se a tarefa precisa de uma pasta.
4. Deixe ligado o "pedir confirmação" antes de enviar, pagar ou apagar.
5. Olhe o histórico de ações do assistente de vez em quando.

Não existe agente imune. Existe agente que não obedece sozinho.

FONTES · conferidas em 24/10/2026

OWASP LLM01:2025 · Prompt Injection
genai.owasp.org/llmrisk/llm01-prompt-injection

Simon Willison · The lethal trifecta (16/06/2025)
simonwillison.net/2025/Jun/16/the-lethal-trifecta

Invariant Labs · GitHub MCP (26/05/2025)
invariantlabs.ai/blog/mcp-github-vulnerability

Aim Security · EchoLeak, CVE-2025-32711 (jun/2025)
Microsoft 365 Copilot · CVSS 9,3 · corrigida

OpenAI Decisions API (beta) · TypeSafe Jev
docs.typesafe.ai · ambos devolvem probabilidade, não decisão

Nikkei (jul/2025) · prompts ocultos em artigos
17 preprints · 14 instituições · 8 países