Guia de Pentest em Projetos de Vibe Coding: Foco nos EUA

Autor: Manus AI **Data:** Junho de 2026

1. Introdução ao Vibe Coding e o Cenário de Segurança nos EUA

O **vibe coding**, uma prática de desenvolvimento de software onde Large Language Models (LLMs) geram código a partir de descrições em linguagem natural, está revolucionando a indústria de tecnologia nos Estados Unidos [1]. Empresas como Microsoft, Google e NVIDIA estão na vanguada da pesquisa e desenvolvimento de IA, impulsionando a adoção de ferramentas de vibe coding como GitHub Copilot, Cursor e Windsurf [2]. Essa aceleração no desenvolvimento, no entanto, introduz um novo e complexo panorama de riscos de segurança, exigindo uma abordagem especializada de *pentesting* [3].

O cenário de segurança cibernética dos EUA, com suas conferências de elite como Black Hat e DEF CON, e suas rigorosas estruturas regulatórias (NIST AI RMF, MITRE ATLAS), tem sido um terreno fértil para a identificação e mitigação dessas novas ameaças. A comunidade de segurança americana tem liderado a pesquisa em *red teaming* de IA, desenvolvendo metodologias e ferramentas para testar a robustez de sistemas de IA e aplicações geradas por ela [4].

Este guia visa fornecer um estudo aprofundado, com foco exclusivo em recursos, ferramentas e expertise dos Estados Unidos, para profissionais de segurança que buscam realizar *pentests* eficazes em projetos de vibe coding.

2. Riscos e Vulnerabilidades Críticas em Vibe Coding (Perspectiva dos EUA)

A proliferação do código gerado por IA, muitas vezes sem a devida revisão humana, tem levado a um aumento de vulnerabilidades. Pesquisas de empresas como Checkmarx e Kaspersky, com forte presença nos EUA, indicam que uma parcela significativa do código gerado por LLMs ainda contém falhas de segurança [5] [6].

2.1. Vulnerabilidades de Aplicação Tradicionais Amplificadas

- **Injeções (SQLi, XSS, Command Injection):** A IA, ao priorizar a funcionalidade, pode gerar código com validação de entrada inadequada, tornando as aplicações suscetíveis a ataques clássicos [6].

- **Segredos Hardcoded:** Chaves de API, credenciais e outros dados sensíveis são frequentemente embutidos diretamente no código gerado, expondo-os a atacantes [6].
- **Controles de Acesso Frágeis:** Lógicas de autenticação e autorização podem ser implementadas de forma insegura (ex: validação apenas no *client-side* ou dependência de cabeçalhos HTTP facilmente manipuláveis) [3].
- **Uso Inseguro de Funções:** A IA pode empregar funções perigosas (como `eval()`) para resolver problemas de forma rápida, criando vetores para execução remota de código (RCE) [6].

2.2. OWASP Top 10 para LLM Applications (Versão 2025)

O projeto OWASP GenAI Security, uma iniciativa global com forte liderança nos EUA, publicou o OWASP Top 10 para LLM Applications, que é o padrão ouro para identificar riscos específicos de IA [7].

ID OWASP	Vulnerabilidade	Descrição e Impacto em Vibe Coding
LLM01	Prompt Injection	Manipulação do LLM através de entradas maliciosas para desviar seu comportamento, ignorar instruções de segurança ou executar ações não intencionais.
LLM02	Sensitive Information Disclosure	O LLM revela inadvertidamente dados confidenciais (PII, segredos, dados de treinamento) em suas respostas, muitas vezes devido a falhas na sanitização de saída ou na gestão de contexto.
LLM03	Supply Chain Vulnerabilities	Dependência de componentes, serviços ou conjuntos de dados comprometidos na cadeia de suprimentos do LLM, levando a envenenamento de dados ou vulnerabilidades no modelo.
LLM04	Data and Model Poisoning	Dados de treinamento manipulados que comprometem a segurança, precisão ou comportamento ético do LLM, resultando em respostas enviesadas ou maliciosas.
LLM05	Improper Output Handling	Falha em validar, sanitizar ou codificar adequadamente as saídas do LLM, podendo levar a ataques *downstream* como XSS, RCE ou injeção de código.
LLM06	Excessive Agency	Concessão de autonomia excessiva a agentes de IA, permitindo que tomem ações não supervisionadas que podem comprometer a confiabilidade, privacidade ou segurança.
LLM07	System Prompt Leakage	Vazamento do *system prompt* do LLM, revelando instruções internas, regras de segurança ou informações confidenciais sobre a arquitetura do sistema.
LLM08	Vector and Embedding Weaknesses	Vulnerabilidades em sistemas de busca de vetores e *embeddings* que podem ser exploradas para ataques de envenenamento de dados, manipulação de contexto ou exfiltração de informações.
LLM09	Misinformation	Geração de informações falsas ou enganosas pelo LLM, comprometendo a tomada de decisões e a credibilidade

ID OWASP	Vulnerabilidade	Descrição e Impacto em Vibe Coding
		da aplicação.
LLM10	Unbounded Consumption	Consumo excessivo de recursos (computacionais, financeiros) devido a *loops* infinitos, prompts complexos ou uso ineficiente do LLM, resultando em negação de serviço ou custos elevados.

2.3. Riscos de Agentes de IA e Model Context Protocol (MCP)

O ecossistema de agentes de IA, com ferramentas como Cursor, Windsurf e Claude Code, e o uso do Model Context Protocol (MCP), introduz novos vetores de ataque. Vulnerabilidades como *Tool Poisoning Attacks* e *Indirect Prompt Injection* via ferramentas ou documentos externos são alvos críticos para *pentesters* [8] [9]. A Microsoft, com sua pesquisa em PyRIT, tem sido uma das líderes na identificação e mitigação desses riscos [10].

---

3. Metodologias de Pentest e Red Teaming de IA (Abordagem dos EUA)

O *red teaming* de IA, conforme praticado por equipes de elite nos EUA (Microsoft, NVIDIA, Google), vai além dos testes de segurança tradicionais, simulando ataques adversariais para descobrir vulnerabilidades antes que os atacantes o façam [4].

3.1. Abordagem Híbrida para Vibe Coding

1. **Reconhecimento e Modelagem de Ameaças:** Compreender a arquitetura da aplicação, identificar os componentes de IA (LLMs, agentes, ferramentas MCP) e mapear os fluxos de dados. Utilizar frameworks como MITRE ATLAS para identificar táticas e técnicas adversariais em sistemas de IA [11].
2. **Análise de Código (SAST) e Componentes (SCA):** Embora o código seja gerado por IA, ferramentas SAST (Static Application Security Testing) e SCA (Software Composition Analysis) são cruciais para identificar vulnerabilidades conhecidas e dependências inseguras. Ferramentas como Checkmarx e Snyk, com recursos aprimorados para IA, são amplamente utilizadas nos EUA [12] [13].
3. **Testes de Aplicação Dinâmicos (DAST) e Interativos (IAST):** Realizar testes em tempo de execução para identificar vulnerabilidades como injeções, falhas de autenticação e

autorização. Ferramentas como Burp Suite, com extensões específicas para LLMs, são indispensáveis [14].

4. **Red Teaming de LLM e Agentes:** Esta é a fase mais crítica para vibe coding. Envolve a simulação de ataques específicos de IA:
    - **Prompt Injection:** Testar a capacidade do LLM de ser manipulado por prompts maliciosos, tanto diretos quanto indiretos (via documentos, e-mails, etc.) [7].
    - **Data Exfiltration:** Tentar extrair dados sensíveis do LLM ou de seus componentes conectados.
    - **Tool Misuse/Excessive Agency:** Se o agente de IA tiver acesso a ferramentas (APIs, sistemas de arquivos), tentar forçá-lo a executar ações não autorizadas ou perigosas [9].
    - **Jailbreaking:** Burlar as salvaguardas do LLM para fazê-lo gerar conteúdo prejudicial ou não permitido.
    - **Model Poisoning:** Simular o envenenamento de dados de treinamento ou de contexto (RAG) para alterar o comportamento do modelo [7].

3.2. Integração Contínua de Segurança (DevSecOps para IA)

As equipes de *red teaming* de IA nos EUA enfatizam a integração da segurança em todo o ciclo de vida de desenvolvimento (DevSecOps). Isso inclui testes automatizados em CI/CD, monitoramento contínuo e a criação de *guardrails* robustos para LLMs [15].

---

4. Arsenal de Ferramentas Americanas para Pentest em Vibe Coding

O mercado americano lidera o desenvolvimento de ferramentas de segurança para IA. As seguintes são consideradas de elite e de “extremo valor” para *pentesters*:

4.1. Ferramentas de Red Teaming e Avaliação de LLMs

- **Promptfoo (Promptfoo Inc., EUA):** Um *framework* focado no desenvolvedor para *red teaming* e avaliação de LLMs. Destaca-se pela capacidade de gerar milhares de ataques específicos para a aplicação, testando *prompt injections*, vazamento de dados e uso indevido de ferramentas. Possui uma interface web intuitiva e integração robusta com CI/CD. Mapeia descobertas para OWASP Top 10 para LLMs, NIST AI RMF e MITRE ATLAS [16] [17].
- **PyRIT (Python Risk Identification Tool) (Microsoft, EUA):** Desenvolvido pela equipe de *AI Red Team* da Microsoft, o PyRIT é um *framework* de código aberto em Python para

orquestrar campanhas adversariais complexas e de múltiplos turnos contra sistemas de IA. É ideal para pesquisadores de segurança que precisam de controle programático e flexibilidade para construir cenários de ataque personalizados [10] [18].

- **Garak (NVIDIA, EUA):** Um scanner de vulnerabilidades de LLM de código aberto mantido pela NVIDIA. Atua como um "Nmap para LLMs", com mais de 120 módulos de *probes* para identificar *prompt injections*, vazamento de dados, *jailbreaks*, desinformação e toxicidade. Focado em testar o modelo em si, em vez da aplicação completa [19] [20].
- **Confident AI (Confident AI, EUA):** Uma plataforma abrangente que combina *red teaming* automatizado, avaliação de LLMs e observabilidade em produção. Oferece mais de 50 vulnerabilidades e 20 vetores de ataque, com mapeamento para OWASP, NIST AI RMF e EU AI Act. É ideal para equipes que buscam uma solução completa para testar e monitorar a segurança de IA em produção [21].

4.2. Ferramentas para Segurança de Agentes e MCP

- **Snyk Agent Scan (Snyk, EUA):** Uma ferramenta CLI para descobrir e escanear componentes de agentes de IA (como Cursor, Windsurf, Claude Code) e servidores MCP na máquina local. Detecta *prompt injections*, envenenamento de ferramentas (*tool poisoning*) e fluxos tóxicos. É crucial para auditar a cadeia de suprimentos de agentes de IA [9] [22].
- **MCP Security Testing (via Promptfoo):** O Promptfoo oferece plugins específicos para testar servidores MCP, simulando ataques de envenenamento de ferramentas e exfiltração de dados, abordando as vulnerabilidades exclusivas introduzidas pelo protocolo [8].

4.3. Ferramentas Tradicionais Adaptadas para IA

- **Burp Suite com Extensões de IA (PortSwigger, Reino Unido - amplamente usado nos EUA):** Embora a PortSwigger seja do Reino Unido, o Burp Suite é uma ferramenta padrão da indústria de *pentest* nos EUA. Extensões como *Burp AI* ou *Atlas AI* (LLM local dentro do Burp) permitem analisar requisições e respostas de APIs que interagem com LLMs, identificando falhas de segurança [14] [23].
- **AI SAST (Static Application Security Testing) (ex: Checkmarx, Snyk, Contrast Security - EUA):** Ferramentas SAST tradicionais que foram aprimoradas com capacidades de IA para detectar padrões de código vulnerável gerado por LLMs com maior precisão e contexto. Empresas americanas como Checkmarx e Snyk são líderes nesse segmento [12] [13] [24].

---

5. Bibliografia, Cursos e Especialistas de Elite dos EUA

Para alcançar um nível de “extremo valor” no *pentest* de vibe coding, é fundamental mergulhar nos recursos e na expertise da comunidade americana.

5.1. Repositórios e Laboratórios Práticos (GitHub - EUA)

- **ottosulin/awesome-ai-security:** Uma lista curada e massiva de *frameworks*, padrões, ferramentas e recursos de aprendizado sobre segurança em IA, com forte inclinação para projetos e pesquisas dos EUA [25].
- **corca-ai/awesome-llm-security:** Focado especificamente em segurança de LLMs, contendo links para os *papers* acadêmicos mais recentes sobre ataques adversariais e defesas, muitos deles de instituições americanas [26].
- **schwartz1375/genai-security-training:** Um currículo de treinamento abrangente e prático para pesquisadores de segurança focado em *red teaming* de sistemas GenAI, com dezenas de *Jupyter Notebooks* para laboratórios práticos. Desenvolvido por um especialista americano [27].
- **LLM Security Labs Playground (llm-sec.dev):** Plataforma interativa com laboratórios práticos baseados no OWASP Top 10 para LLMs, permitindo experimentação segura com vulnerabilidades [28].
- **microsoft/AI-Red-Teaming-Playground-Labs:** Repositório da Microsoft com *Jupyter Notebooks* que demonstram o uso do PyRIT para resolver desafios de *red teaming* de IA [29].

5.2. Cursos e Treinamentos de Elite (EUA)

- **SANS Institute (EUA):** Reconhecido globalmente, o SANS oferece cursos de ponta em segurança cibernética. Destacam-se:
  - **SEC535: Offensive AI - Attack Tools and Techniques:** Focado em alavancar ferramentas de IA para operações ofensivas e testar a segurança de sistemas de IA [30].
  - **SEC545: GenAI and LLM Application Security:** Aborda a segurança de aplicações GenAI e LLM, incluindo vulnerabilidades e mitigações [31].
- **Hack the AI Agent (GitHub Secure Code Game):** Um jogo de código aberto gratuito do GitHub (empresa americana) focado em encontrar e explorar vulnerabilidades do mundo real em IA agêntica [32].
- **Microsoft AI Red Teaming 101 Series:** Uma série de treinamentos da Microsoft que aborda vulnerabilidades, técnicas de ataque e estratégias de defesa para sistemas de IA

generativa [33].

5.3. Artigos, Relatórios e Conferências de Valor Extremo (EUA)

- **Black Hat USA e DEF CON (Las Vegas, EUA):** As conferências mais prestigiadas de segurança cibernética, onde as últimas pesquisas e *exploits* em segurança de IA são apresentados. Acompanhar as palestras e *workshops* dessas conferências é fundamental para se manter atualizado [34] [35].
  - **Backslash Security na Black Hat USA 2025:** Anunciou uma plataforma abrangente de segurança para vibe coding, destacando a importância da visibilidade e governança em infraestruturas de código AI [36].
  - **DEF CON AI Village:** Um espaço dedicado na DEF CON para discussões e demonstrações práticas sobre segurança e *red teaming* de IA [37].
- **Relatórios de Pesquisa de Big Techs (Microsoft, NVIDIA, Google, Anthropic):** Essas empresas publicam regularmente pesquisas de ponta sobre segurança de IA, *red teaming* e mitigação de riscos. O *AI Red Team* da NVIDIA, por exemplo, é uma referência na área [2] [4].
- **Stanford HAI (Human-Centered Artificial Intelligence) (Stanford University, EUA):** Publica o *AI Index Report* anualmente, que inclui seções dedicadas à segurança e responsabilidade da IA, além de pesquisas acadêmicas de alto impacto [38].
- **MITRE ATLAS (MITRE Corporation, EUA):** Uma base de conhecimento pública de táticas e técnicas adversariais baseadas em observações de ataques reais e demonstrações de *red teams* de IA. Essencial para modelagem de ameaças [11].
- **Cloud Security Alliance (CSA) (EUA):** Publica guias e *frameworks* sobre segurança de IA, incluindo o “Secure Vibe Coding Guide”, que oferece diretrizes práticas para o desenvolvimento seguro [1].

5.4. Especialistas e Influenciadores para Acompanhar (EUA)

- **Equipes de AI Red Team da Microsoft, NVIDIA e Google:** Seguir os pesquisadores e engenheiros dessas equipes no LinkedIn e Twitter para obter *insights* sobre as últimas tendências e descobertas.
- **Caleb Sima (Co-host do AI Security Podcast):** Ex-CISO e especialista em segurança de IA, oferece análises aprofundadas sobre o cenário de ameaças [39].
- **Ashish Rajan (Co-host do AI Security Podcast):** Outro ex-CISO e especialista, conhecido por suas discussões sobre segurança de IA para líderes e CISOs [39].

- **Ram Shankar Siva Kumar (Microsoft):** Especialista em *AI Red Teaming* e autor de publicações influentes na área [4].
- **Ken Huang (CSA Fellow):** Co-Chair do grupo de trabalho de segurança em IA da Cloud Security Alliance, autor de guias sobre vibe coding seguro [1].
- **Influenciadores em AI/ML Security no LinkedIn e Twitter:** Profissionais como Andrew Ng, Allie K. Miller e outros líderes de pensamento que compartilham conteúdo valioso sobre as últimas tendências e desafios em segurança de IA [40].

Referências

[1] Cloud Security Alliance. “Secure Vibe Coding Guide”. Disponível em: https://cloudsecurityalliance.org/blog/2025/04/09/secure-vibe-coding-guide [2] NVIDIA. “NVIDIA AI Red Team: An Introduction”. Disponível em: https://developer.nvidia.com/blog/nvidia-ai-red-team-an-introduction/ [3] NetSPI. “Vibe Coding: A Pentester’s Dream”. Disponível em: https://www.netspi.com/blog/executive-blog/web-application-pentesting/vibe-coding-a-pentesters-dream/ [4] Stanford HAI. “A Few Useful Lessons about AI Red Teaming”. Disponível em: https://hai.stanford.edu/events/ram-shankar-siva-kumar-few-useful-lessons-about-ai-red-teaming [5] Checkmarx. “Vibe Coding Security: Risks, Vulnerabilities, and Secure AI Coding”. Disponível em: https://checkmarx.com/blog/security-in-vibe-coding/ [6] Kaspersky. “Security risks of vibe coding and LLM assistants for developers”. Disponível em: https://www.kaspersky.com/blog/vibe-coding-2025-risks/54584/ [7] OWASP. “OWASP Top 10 for Large Language Model Applications”. Disponível em: https://genai.owasp.org/llm-top-10/ [8] Promptfoo. “MCP Security Testing Guide”. Disponível em: https://www.promptfoo.dev/docs/red-team/mcp-security-testing/ [9] Snyk. “agent-scan: Security scanner for AI”. Disponível em: https://github.com/snyk/agent-scan [10] Microsoft Azure. “Python Risk Identification Tool for generative AI (PyRIT)”. Disponível em: https://github.com/Azure/PyRIT [11] MITRE. “MITRE ATLAS”. Disponível em: https://atlas.mitre.org/ [12] Checkmarx. “Best AI Cybersecurity Tools in 2026”. Disponível em: https://checkmarx.com/learn/ai-security/best-ai-cybersecurity-tools-top-9-to-watch-in-2026/ [13] Wiz. “AI SAST: Smarter Static Application Security Testing”. Disponível em: https://www.wiz.io/academy/application-security/ai-sast [14] PortSwigger. “Web LLM attacks”. Disponível em: https://portswigger.net/web-security/llm-attacks [15] Microsoft. “AI red teaming training series: securing generative AI systems”. Disponível em: https://learn.microsoft.com/en-us/security/ai-red-team/training [16] Promptfoo. “Top Open Source AI Red-Teaming and Fuzzing Tools in 2025”. Disponível em: https://www.promptfoo.dev/blog/top-5-open-source-ai-red-teaming-tools-2025/ [17] Promptfoo. “Promptfoo: LLM evals & red teaming”. Disponível em: https://github.com/promptfoo/promptfoo [18] Promptfoo. “Promptfoo vs PyRIT: A Practical Comparison of LLM Red Teaming Tools”. Disponível em: https://www.promptfoo.dev/blog/promptfoo-vs-pyrit/ [19] NVIDIA. “garak: the LLM vulnerability scanner”. Disponível em: https://github.com/NVIDIA/garak [20] AppSec Santa.

“Garak vs Promptfoo (2026): LLM Security Testing”. Disponível em: https://appsecsanta.com/ai-security-tools/garak-vs-promptfoo [21] Confident AI. “5 Best AI Red Teaming Tools to Find LLM Vulnerabilities in 2026”. Disponível em: https://www.confident-ai.com/knowledge-base/compare/best-ai-red-teaming-tools-2026 [22] Snyk. “Agent Scan: Discover and scan agent components on your machine for prompt injections and vulnerabilities”. Disponível em: https://github.com/snyk/agent-scan [23] CovertSwarm. “Atlas AI: Local LLM inside Burp Suite”. Disponível em: https://www.covertswarm.com/post/atlas-ai-local-ai-plugin [24] Contrast Security. “DAST vs AI Code: Why Dynamic Application Security Testing Can’t Keep Pace”. Disponível em: https://www.contrastsecurity.com/security-influencers/dast-vs-ai-code-why-dynamic-application-security-testing-cant-keep-pace [25] Otto Sulin. “awesome-ai-security”. Disponível em: https://github.com/ottosulin/awesome-ai-security [26] Corca AI. “awesome-llm-security”. Disponível em: https://github.com/corca-ai/awesome-llm-security [27] Schwartz1375. “genai-security-training”. Disponível em: https://github.com/schwartz1375/genai-security-training [28] LLM Security Labs. “Interactive LLM Security Labs Playground”. Disponível em: https://www.llm-sec.dev/ [29] Microsoft. “AI-Red-Teaming-Playground-Labs”. Disponível em: https://github.com/microsoft/AI-Red-Teaming-Playground-Labs [30] SANS Institute. “SEC535: Offensive AI - Attack Tools and Techniques”. Disponível em: https://www.sans.org/cyber-security-courses/offensive-ai-attack-tools-techniques [31] SANS Institute. “SEC545: GenAI and LLM Application Security”. Disponível em: https://www.sans.org/cyber-security-courses/genai-llm-application-security/ [32] GitHub. “Hack the AI agent: Build agentic AI security skills with the GitHub Secure Code Game”. Disponível em: https://github.blog/security/hack-the-ai-agent-build-agentic-ai-security-skills-with-the-github-secure-code-game/ [33] Microsoft. “AI Red Teaming 101 Series”. Disponível em: https://learn.microsoft.com/en-us/security/ai-red-team/training [34] Black Hat. “Black Hat USA”. Disponível em: https://www.blackhat.com/us-26/ [35] DEF CON. “DEF CON® Hacking Conference Home”. Disponível em: https://defcon.org/ [36] Backslash Security. “Backslash Security to Unveil Comprehensive Vibe Coding Security Platform at Black Hat USA 2025”. Disponível em: https://www.backslash.security/press-releases/backslash-security-to-unveil-comprehensive-vibe-coding-security-platform-at-black-hat-usa-2025 [37] AI Village. “DEF CON AI Village”. Disponível em: https://aivillage.org/ [38] Stanford HAI. “The 2026 AI Index Report”. Disponível em: https://hai.stanford.edu/ai-index/2026-ai-index-report [39] AI Security Podcast. “AI Security Podcast”. Disponível em: https://open.spotify.com/show/3nV4eijfzdHKIvDOaycVII [40] DataNorth AI. “Top 10 AI influencers to follow on LinkedIn in 2026”. Disponível em: https://datanorth.ai/blog/top-10-ai-influencers-to-follow-on-linkedin-in-2026