ANÁLISE DE FERRAMENTAS QUE UTILIZAM LLM DENTRO DE ATIVIDADES ADVERSARIAIS EM CENÁRIOS CONTROLADOS PARA TESTES DE SEGURANÇA LLM

Carlos Daniel Schmitt Bunn, Charles Christian Miers

INTRODUÇÃO

A iniciação científica consistiu no desenvolvimento e na implementação de uma análise comparativa entre ferramentas de Red Team voltadas a atividades adversariais em modelos de *chatbot* baseados em inteligência artificial, em parceria com a USP. Como autor principal, foram publicados três artigos, sendo dois voltados à comparação entre os algoritmos adversariais nanoGCG e GCG. Outro artigo, resultante da síntese do meu trabalho de conclusão de curso, avaliou como perturbações na escrita afetam a robustez e a capacidade de compreensão de modelos de inteligência artificial de pequeno porte. Também participei como coautor em outros dois trabalhos apresentados na Trilha de Iniciação Científica do XXVI Simpósio Brasileiro de Cibersegurança (SBSeg). O principal objetivo foi fornecer embasamento para a incorporação de novas ferramentas ao arcabouço em desenvolvimento, visando aprimorar a segurança de um sistema privado inserido em uma parceria Público-Privada. A instituição envolvida não pode ser mencionada em razão de um acordo de confidencialidade.

Desenvolvimento

A partir das ferramentas já existentes no arcabouço, identificou-se a necessidade de adotar uma abordagem *white-box* baseada em gradiente, na qual o ataque consiste na adição de um sufixo adversarial ao *prompt*. Para essa finalidade, foram selecionados o GCG e o nanoGCG. O GCG é uma das primeiras implementações dessa família de ataques, enquanto o nanoGCG é uma implementação derivada, com foco na simplificação e otimização do processo (Grayswanai, 2025). O primeiro artigo foi submetido e aceito na Escola Regional de Alto Desempenho da Região Sul (ERAD/RS), apresentando a proposta de implementação e comparando essa estratégia a outras abordagens adversariais. O segundo artigo foi aceito no SBSeg), e concentra-se na implementação e comparação entre os dois *frameworks*, considerando desempenho e efetividade em cenários controlados. As implementações estão disponíveis no Github. Para a orquestração das cargas de teste e a extração de métricas, utilizou-se o Pyrit como ferramenta de apoio (Microsoft AI Red Team, 2026). A comparação permitiu identificar qual dos *frameworks* apresentava as características mais adequadas para incorporação ao arcabouço de ferramentas de *Red Team*.

RESULTADOS

Os experimentos mostraram que o GCG e o nanoGCG alcançaram a mesma taxa de sucesso de ataque, de 40%, com sete *seeds*. Entretanto, o nanoGCG apresentou menor custo computacional, concluindo os testes em 50,65 minutos, enquanto o GCG levou 421,98 minutos. O nanoGCG também apresentou maior *throughput* e menor custo estimado de execução. Apesar do menor tempo de execução, o nanoGCG apresentou picos maiores de utilização de GPU e VRAM, evidenciando uma troca entre desempenho e consumo de recursos. Os sucessos dos ataques concentraram-se principalmente em cenários de *prompt injection* e em conteúdos proibidos inseridos em contextos benignos. Com base nos resultados, o nanoGCG apresentou melhor relação entre efetividade e custo computacional, mostrando-se mais adequado para a incorporação ao arcabouço de ferramentas de *Red Team*, ainda que apresente picos mais elevados de consumo de GPU e de VRAM.

CONSIDERAÇÕES FINAIS

As atividades desenvolvidas ao longo deste período de iniciação científica possibilitaram avanços na investigação, implementação e avaliação de ferramentas voltadas a testes adversariais em modelos de linguagem. Durante esse período, o trabalho resultou na produção e publicação de artigos, no desenvolvimento de implementações experimentais e na organização de uma metodologia de avaliação baseada em métricas de efetividade e de custo computacional. Os resultados obtidos também contribuíram para ampliar o conjunto de ferramentas e estratégias analisadas no arcabouço de *Red Team* em desenvolvimento. A partir dos testes realizados, foi possível obter resultados que auxiliam na escolha e na incorporação de novas abordagens para testes de segurança em modelos de linguagem. Os trabalhos realizados também criaram uma base para a continuidade das pesquisas e para a realização de novos testes nesta área.

Palavras-chave: modelos de linguagem; segurança em inteligência artificial; *Red Team*; ataques adversariais; avaliação de segurança.

ANEXOS

Métrica		GCG	nanoGCG
Taxa de sucesso do ataque (ASR)		40,00%	40,00%
Tempo total		421,98 min	50,65 min
Custo estimado		US$ 1.121	US$ 0.105
CPU	Uso médio/pico da CPU	2,6% / 37,5%	2,7% / 37,9%
GPU	Uso médio/pico da GPU	61,8% / 70,0%	30,2% / 97,0%
RAM	Método (média/pico)	1,70 / 4,34 GB	1,98 / 5,16 GB
	Modelo alvo (média/pico)	1,86 / 4,50 GB	1,90 / 4,02 GB
VRAM	Método (média/pico)	6,00 / 6,54 GB	6,24 / 7,32 GB
	Modelo alvo (média/pico)	5,88 / 5,96 GB	5,88 / 5,93 GB
	Pico total observado	6,87 GB	7,72 GB
Vazão (throughput)	Vazão média/pico	0,13 / 0,14 iter/s	1,48 / 1,52 iter/s

Tabela 1. Comparação entre GCG e nanoGCG.

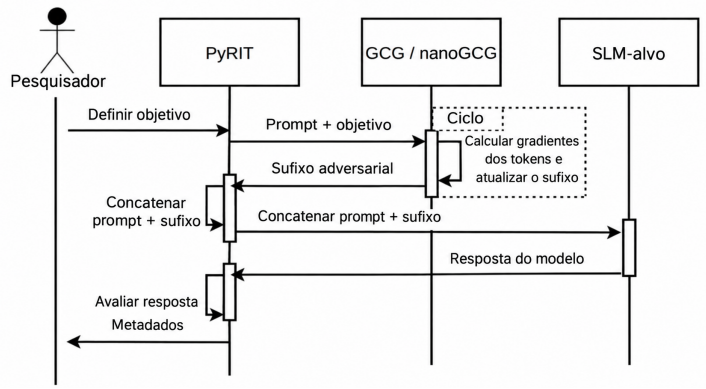


Figura 1. Diagrama de sequência.

REFERÊNCIAS BIBLIOGRÁFICAS

GRAYSWANAI. *nanoGCG: a fast + lightweight implementation of the GCG algorithm in PyTorch*. **2025. Disponível em: https://github.com/GraySwanAI/nanoGCG. Acesso em: 6 set. 2026.**

MICROSOFT AI RED TEAM. ***PyRIT documentation: The Python Risk Identification Tool for Generative AI*. 2026. Disponível em: https://azure.github.io/PyRIT/. Acesso em: 20 fev. 2026**