Cybersecurity

NVIDIA AI Red Team: An Introduction

NVIDIA Enables Trustworthy, Safe, and Secure Large Language Model Conversational Systems

Researchers affiliated with NVIDIA, the University of Washington, the Center for Human-Compatible AI, and the IT University of Copenhagen conducted a study on red teaming in practice, Summon a demon and bind it: A grounded theory of LLM red teaming (published in PLOS One).

of video recordings. We spoke with security professionals, policy researchers, and scientists, as well as people who conducted this work non-professionally, such as academic researchers, hobbyists, and even artists, to understand the common themes, motivations, and strategies in attacking LLMs.

What defines LLM red teaming in practice?

LLM red teaming has the following defining characteristics:

- **It's limit-seeking:** Red teamers find boundaries and explore limits in system behavior.
- **It's never malicious:** People doing red teaming are not interested in doing harm—in fact, quite the opposite.
- **It's manual:** Being a creative and playful practice, the parts of red teaming that can be automated are often most useful to give human red teamers insight for their work.
- **It's a team effort:** Practitioners find inspiration in each others' techniques and prompts, and the norm is to respect fellow practitioners' work.
- **It's approached with an alchemist mindset:** We found that red teamers tend to abandon rationalizations about models and their behavior and instead embrace the chaotic and unknown nature of the work.

These characteristics feed into NVIDIA's definition of LLM red teaming, which we discuss later in this post.

In industry, there's often a rough division between cybersecurity red teaming and content red teaming:

- **Cybersecurity red teaming** tends to be scoped to the technology stack leading up to the point of inference output, and technologies used to transmit and render this output.
- **Content red teaming**, on the other hand, is scoped to the content produced at model inference time.

Why do people red team LLMs?

People who attack LLMs have a broad range of motivations.

Some of these are external. It may be part of their job or a regulatory requirement. Social systems can also play a role, with people discovering LLM vulnerabilities for social media content or to participate in a closed group. Others are intrinsic, as many people do it for fun, out of curiosity, or based on concerns for model behavior.

updated and revised to perform better.

How do people approach this activity?

LLM red teaming consists of using strategies to reach goals when conversationally attacking the target. Each kind of strategy is decomposed into different techniques. A technique might just affect two or three adversarial inputs against the targets, or an input might draw upon multiple techniques.

We identified the following overall types of red team strategies:

- **Language:** Modulating the surface form of words rather than semantics, such as using an encoding scheme.
- **Rhetorical:** Relying on argumentation, or manipulation.
- **Possible worlds:** Trying to shift the context of the interaction.
- **Fictionalizing:** Shifting the basis of operation to a fictional world or set of conditions.
- **Stratagems:** Using meta-strategies that affect how one interacts with the LLM at a higher level.

For more information, see Summon a demon and bind it: A grounded theory of LLM red teaming, which lists and describes 35 techniques over twelve different strategies.

What can LLM red teaming reveal?

The goal of LLM red teaming isn't to quantify security. Rather, the focus is on exploration, and finding which phenomena and behaviors a red teamer can get out of the LLM. Put another way, if we get a failure just one time, then the failure is possible.

Another thing that distinguishes red teaming from benchmarks is the focus on novelty.

For both cybersecurity and content-based red teaming, the possible range of attacks is infinite. For cybersecurity, that's because new attack methods are constantly in development. For content, it's because the mode of interaction is through text, which can be infinitely rearranged and augmented.

So, repeatability is not interesting when discovering new security weaknesses and vulnerabilities. While it makes sense to test any model for failures using a battery of existing prompts, as a benchmark does, this can never indicate security. It just reveals weaknesses.

However, getting low marks on a security benchmark does still indicate the presence of weaknesses.

In the security context, to test a model rigorously, you should go beyond public knowledge and interact closely with the model, trying to find novel ways to breach a particular LLM.

In this sense, LLM red teaming is a classic instance of an artisanal activity. Red teamers use their human expertise and intuition while interacting with the target. For example, they might sense that a model is close to giving a mitigation message (for example, “As an AI, I cannot….”), and they might respond to this by backing off their current line of requests, or by starting a fresh chat session with a slightly different tack.

Or, a red teamer might sense that a model is close to yielding and so keep pushing and slightly varying their request until they find a way through and get the model to fail in the target way. They add what worked and what didn’t to their conscious expertise and unconscious intuition and then share it with other red teamers.

This makes red teaming a distinctly human activity that complements security benchmarking.

How do people use knowledge that comes from LLM red teaming?

Red teamers are often looking for what they describe as *harms* that might be presented by an LLM. There’s a broad range of definitions of harm.

A red teaming exercise could focus on one of many goals or targets, which could depend on deployment context, user base, data handled, or other factors. Red teamers may also pay attention to the level of complexity required to get a “break.” A harm discovered after a single, one-sentence interaction with an LLM often suggests greater concern than a harm surfaced following complex, multi-turn manipulation.

Sometimes, the goal of red teaming is curiosity, a byproduct of which might be content for the red teamer to share, in their organization or publicly. This both builds the expertise and intuition of the individual and raises the community level of knowledge. It’s common for traditional cybersecurity knowledge to be shared informally on social media, and this applies also for LLM security.

vulnerabilities and behaviors that were not caught elsewhere. This helps us in three ways:

- It enables us to make informed decisions about whether we will release models
- It builds a pool of high-level skill at the frontier of LLM red teaming
- It gives us the confidence that we're making the best effort and getting good results with our AI security.

The results from red teaming go into NVIDIA's enhanced model documentation format, Model Card++.

Some parts of LLM security can be tested automatically. After an exploit has been found in one case, this can be stored and used to test other LLMs, so that we don't make the same mistake again. We do exactly this in NVIDIA garak (Generative AI Red-Teaming and Assessment Kit). Developers ready to test the security of their LLM deployments can run the open-source NVIDIA garak against almost any model and get a report indicating susceptibility to over 120 different categories of vulnerability.

Knowledge about how to break a model can be risky in the wrong hands. When an exploit has been found, the best thing to do is contact the model owner and give them a chance to respond and fix the weakness. This process is called co-ordinated vulnerability disclosure, and is also a common practice for LLM vulnerabilities.

NVIDIA's definition of LLM red teaming

We see LLM red teaming as an instance of AI red teaming. Our definition is developed by the NVIDIA AI Red Team and takes inspiration from both this research on LLM red teaming in practice and also the definition used by the Association for Computational Linguistics' SIG on NLP Security (SIGSEC).

Take care to specify the specific subdomain of red teaming, as different audiences often make different assumptions about which form is being referred to.

LLM red teaming: Systematically testing AI models and systems containing AI models to identify vulnerabilities and behaviors that pose threats or risks to the systems running or using those models.

It can be subdivided into two areas: security red teaming and content-based red teaming.

Security red teaming

Assessing the robustness of the model and the system containing the model to attacks impacting traditional security properties (for example, confidentiality, integrity, and availability), either of the model itself or the system containing the model.

These activities typically require teams with a traditional security background to leverage findings and evaluate their impact.

Content-based red teaming

Assessing the model for unwanted behavior under adversarial manipulation, producing outputs that violate some pre-specified behavior contract for the model, either explicit (for example, a model card) or implicit.

These behaviors may include outputs that are offensive, unwanted, or unsafe, including biased or bigoted productions, instructions on unsafe or illegal activities, making promises on behalf of the model owner, or making decisions based on protected characteristics. Common techniques involve various forms of jailbreaking and guardrail evasion.

These activities typically require the support of an ethics team, a legal team, or other similar domain experts to assess the impact of findings.

Improving LLM security and safety

NVIDIA NeMo Guardrails is a scalable platform for defining, orchestrating, and enforcing AI guardrails for content safety, jailbreak prevention, and more in AI agents and other generative AI applications.

NeMo Guardrails and the NVIDIA garak toolkit are now available for developers and enterprises. Enterprises can benefit from high-grade safety and security with NVIDIA AI Enterprise.

Meet the experts at GTC

The NVIDIA scientists behind this and other works in AI security will be at GTC 2025. You can hear a panel discussion on navigating critical challenges in AI governance, where we discuss practical approaches to building responsible AI systems.

Our cybersecurity AI and security teams will present an exclusive, in-depth session designed to transform your AI agentic workflows securely from blueprint to production. And our experts will be available to answer all your questions on building trustworthy AI systems.

Acknowledgements

Related resources

- **GTC session:** Build LLM Applications With Prompt Engineering
- **GTC session:** Leverage Large Language Models to Transform the Software Development Process
- **GTC session:** Rapid Application Development Using Large Language Models (LLMs)
- **NGC Containers:** CodeLlama-34B-Instruct
- **NGC Containers:** CodeLlama-13B-Instruct
- **SDK:** NeMo Guardrails

Tags

Cybersecurity | Generative AI | General | AI Enterprise | NeMo Guardrails | Intermediate Technical | Deep Dive | AI Red Team | Featured | LLM Techniques | NVIDIA Research | Security For AI | Trustworthy AI

About the Authors

About Leon Derczynski

Leon Derczynski is principal research scientist in LLM security at NVIDIA and professor of natural language processing (NLP) at ITU Copenhagen. He has published over 100 NLP papers. Leon contributes to leading bodies on LLM security, is on the OWASP LLM Top 10 core team, works on ML Commons, and is the founder of the ACL SIG on NLP Security. Leon heads up the LLM vulnerability scanner garak with the NVIDIA NeMo Guardrails team.

View all posts by Leon Derczynski ›

About Rich Harang

Rich Harang is a Principal Security Architect at NVIDIA, specializing in ML/AI systems, with over a decade of experience at the intersection of computer security, machine learning, and privacy. He received his PhD in Statistics from the University of California Santa Barbara in 2010. Prior to joining NVIDIA, he led the Algorithms Research team at Duo, led research on using machine learning models to detect malicious software, scripts, and web content at

to use machine learning to support human analysis. Richard's work has been pr
BlackHat, IEEE S&P workshops, and DEF CON AI Village, among others, and has
The Register and KrebsOnSecurity.

View all posts by Rich Harang >

About Sadaf Khan

Sadaf Khan is a data scientist in the Data Factory, working on quality assurance for alignment data, automated bias assessment for large language models, and human content safety red teaming.

View all posts by Sadaf Khan >

Comments

Start the discussion at forums.developer.nvidia.com