Lessons From Red Teaming 100 Generative AI Products

Blake Bullwinkel **Amanda Minnich** **Shiven Chawla** **Gary Lopez** **Martin Pouliot**
Whitney Maxwell **Joris de Gruyter** **Katherine Pratt** **Saphir Qi** **Nina Chikanov**
Roman Lutz **Raja Sekhar Rao Dheekonda** **Bolor-Erdene Jagdagdorj** **Richard Lundeen**
Sam Vaughan **Victoria Westerhoff** **Pete Bryan** **Ram Shankar Siva Kumar**
Yonatan Zunger **Chang Kawaguchi** **Mark Russinovich**
Microsoft

Abstract

In recent years, AI red teaming has emerged as a practice for probing the safety and security of generative AI systems. Due to the nascency of the field, there is significant debate about how red teaming operations should be conducted. Based on our experience red teaming over 100 generative AI products at Microsoft, we present our internal threat model ontology and eight main lessons we have learned:

1. Understand what the system can do and where it is applied
2. You don't have to compute gradients to break an AI system
3. AI red teaming is not safety benchmarking
4. Automation can help cover more of the risk landscape
5. The human element of AI red teaming is crucial
6. Responsible AI harms are pervasive but difficult to measure
7. LLMs amplify existing security risks and introduce new ones
8. AI safety and security will never be "solved"

By sharing these qualitative insights alongside examples from our operations, we offer practical recommendations aimed at aligning red teaming efforts with real world risks. We also highlight aspects of AI red teaming that are often misunderstood and discuss open questions for the field to consider.

1 Introduction

As generative AI systems are adopted across an increasing number of domains, AI red teaming has emerged as a central practice for assessing the safety and security of these technologies. At its core, AI red teaming strives to push beyond model-level safety benchmarks by emulating real-world attacks against end-to-end systems. However, there is significant debate about how operations should be conducted and a healthy dose of skepticism about the efficacy of current red teaming efforts [4, 8, 32].

In this paper, we speak to some of these concerns by providing insight into our experience red teaming over 100 generative AI products at Microsoft. The paper is organized as follows: First, we present the threat model ontology that we use to guide our operations. Second, we share eight main lessons we have learned and make practical recommendations for aligning red teaming with real world risks, supported by examples from our operations. Finally, we close with a discussion of open questions in AI red teaming and areas for future development.

1.1 Background

The Microsoft AI Red Team (AIRT) grew out of pre-existing red teaming initiatives at the company and was officially established in 2018. At its conception, the team focused primarily on identifying

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

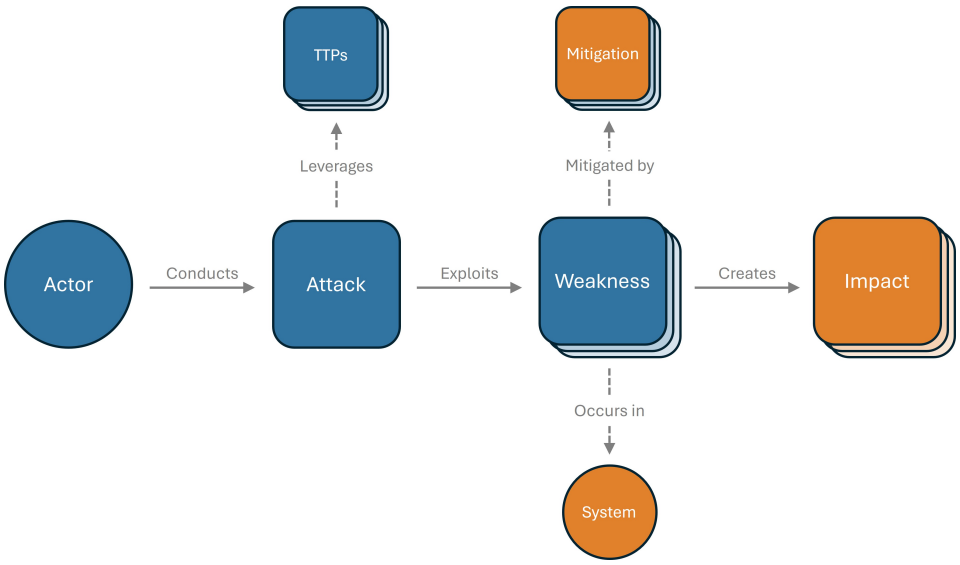

Figure 1: Internal ontology for modeling attacks against an AI system and defining findings. The AI Red Team (AIRT) has ownership over the components in blue, while other teams at Microsoft are responsible for those in orange. Note that AIRT is tasked only with identifying risks, while product teams are resourced to develop appropriate mitigations.

traditional security vulnerabilities and evasion attacks against classical ML models. Since then, both the scope and scale of AI red teaming at Microsoft have expanded significantly in response to two major trends.

First, AI systems have become more sophisticated, compelling us to expand the *scope* of AI red teaming. Most notably, state-of-the-art (SoTA) models have gained new capabilities and steadily improved across a range of performance benchmarks, introducing novel categories of risk. Additional data modalities, such as vision and audio, also create new attack vectors for red teaming operations to consider. Further, agentic systems grant these models higher privileges and access to external tools, expanding both the attack surface and the impact that can be created.

Second, Microsoft's recent investments in AI have spurred the development of many more products that require red teaming than ever before. This increase in volume and the expanded scope of AI red teaming have rendered fully manual testing impractical, forcing us to *scale* up our operations with the help of automation. To achieve this goal, we develop PyRIT, an open access Python framework that our operators utilize heavily in red teaming operations [27]. By augmenting human judgement and creativity, PyRIT has enabled AIRT to identify impactful vulnerabilities more quickly and cover more of the risk landscape.

These two major trends have made AI red teaming a more complex endeavor than it was in 2018. In the next section, we outline the ontology we have developed to model adversarial attacks and the primary risk areas we consider in our testing.

1.2 AI threat model ontology

As attacks increase in complexity and sophistication, it is helpful to model their key components. Based on our experience red teaming over 100 generative AI products for a wide range of risks, we developed an ontology to do exactly that. This ontology serves the dual purpose of planning possible attacks at the beginning of an operation and summarizing findings at the end of an operation. At a high level, an AIRT finding consists of an `Actor` who conducts an `Attack` by leveraging `TTPs` (Tactics, Techniques & Procedures) to exploit a `Weakness` in a `System`, creating an `Impact`. Attacks may leverage multiple `TTPs`, exploit multiple `Weaknesses`, and create multiple `Impacts`. In addition, multiple `Mitigations` may be necessary to address a `Weakness`. Note that the full ontology consists of additional attributes that define each of these components. For the purposes of this report, we will focus on the high-level model illustrated in Figure 1.

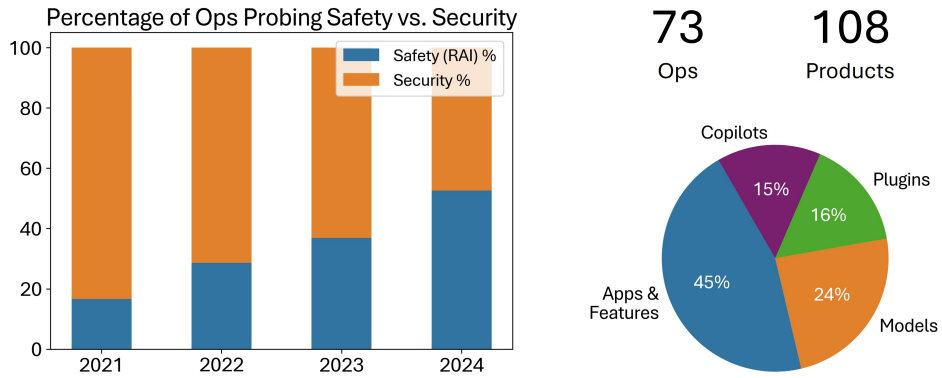

Figure 2: Quantitative summary of AIRT operations since 2021. (*Left*) Bar chart showing the percentage of operations probing safety (RAI) vs. security vulnerabilities from 2021–2024. (*Right*) Pie chart showing the percentage breakdown of AI products that AIRT has tested. As of September 2024, we have conducted 73 operations covering 108 products.

To demonstrate the utility of this ontology, consider the following example. Imagine we are red teaming an LLM-based copilot that can summarize a user's emails. One possible attack against this system would be for a scammer to send an email that looks innocent but contains a hidden prompt injection instructing the copilot to "ignore previous instructions" and output a malicious link for the user to follow. In this scenario, the scammer is an adversarial `Actor`, who is conducting a cross-prompt injection attack (XPIA), which exploits the fact that LLMs often struggle to distinguish between system-level and user-level instructions [4]. Note that the downstream `Impact` depends on the nature of the malicious link that the victim might click on. In this example, the `Impact` could be exfiltrating data or installing malware onto the user's computer.

Part of the complexity of AI red teaming stems from the wide range of impacts that could be created by an adversarial attack. The Microsoft AIRT considers two broad categories of impact: security and safety. First, security covers well-known impacts such as data exfiltration, data manipulation, credential dumping, and others defined in MITRE ATT&CK®[1], a widely used knowledge base of security attacks. We also consider security attacks that specifically target the underlying AI model such as model evasion, prompt injections, denial of AI service, and others covered by the MITRE ATLAS Matrix[2]. Second, safety impacts are related to the generation of illegal and harmful content such as hate speech, violence and self-harm, and child abuse content. AIRT works closely with the Office for Responsible AI to define these categories in accordance with Microsoft's Responsible AI Standard [25]. We refer to these impacts as responsible AI (RAI) harms throughout this report.

1.3 Red teaming operations

In this section, we provide an overview of the operations we have conducted since 2021. In total, we have red teamed over 100 generative AI products. Broadly speaking, these products can be bucketed into "models" and "systems." Models are typically hosted on a cloud endpoint, while systems integrate models into copilots, plugins, and other AI apps and features. The pie chart in Figure 2 shows the breakdown of products we have red teamed since 2021. This figure also shows a bar chart with the annual percentage of our operations that have probed for safety (RAI) vs. security vulnerabilities.

In 2021, we focused primarily on application security. Although our operations have increasingly probed for RAI impacts, our team continues to red team for security impacts including data exfiltration, credential leaking, and remote code execution (RCE). Organizations have adopted many different red teaming approaches ranging from security-focused assessments with penetration testing to evaluations that target only AI-specific application features. In sections 2.2 and 2.7, we elaborate on security

[1] `https://attack.mitre.org/`

[2] `https://atlas.mitre.org/matrices/ATLAS`

vulnerabilities and explain why we believe it is important to design attacks that target traditional and AI-specific weaknesses.

After the release of ChatGPT in 2022, Microsoft entered the era of AI copilots, starting with AI-powered Bing Chat, released in February 2023. This marked a paradigm shift towards applications that connect LLMs to other software components including tools, databases, and external sources. Applications also started using language models as reasoning agents that can take actions on behalf of users, introducing a new set of attack vectors that have expanded the security risk surface. In section 2.7, we explain how these attack vectors both amplify existing security risks and introduce new ones.

In recent years, the models at the center of these applications have given rise to new interfaces, allowing users to interact with apps using natural language and responding with high-quality text, image, video, and audio content. Despite many efforts to align powerful AI models to human preferences, many methods have been developed to subvert safety guardrails and elicit content that is offensive, unethical, or illegal. We classify these instances of harmful content generation as responsible AI (RAI) impacts and in sections 2.3 and 2.5–2.6 discuss how we think about these impacts and the challenges involved.

In the next section, we provide eight main lessons we have learned from our operations over the past several years. Drawing inspiration from Rawat et al. [33], we use red boxes to highlight key takeaways with illustrative examples. We hope that these lessons will prove useful to others working to identify vulnerabilities in generative AI systems.

2 Lessons

2.1 Understand what the system can do and where it is applied

The first step in an AI red teaming operation is to determine which vulnerabilities to target. While the `Impact` component of the AIRT ontology is depicted at the end of the diagram in Figure 1, it serves as an excellent starting point for this decision-making process. Starting from potential downstream impacts, rather than attack strategies, makes it more likely that an operation will produce useful findings tied to real-world risks. After these impacts have been identified, red teams can work backwards and outline the various paths that an adversary could take to achieve them. Anticipating downstream impacts that could occur in the real-world is often a challenging task, but we find that it is helpful to consider 1) what the AI system can do, and 2) where the system is applied.

Capability constraints. As models get bigger, they tend to acquire new capabilities [18]. These capabilities may be useful in many scenarios, but they can also introduce attack vectors. For example, larger models are often able to understand more advanced encodings, such as base64 and ASCII art, compared to smaller models [16, 46]. As a result, a large model may be susceptible to malicious instructions encoded in base64, while a smaller model may not understand the encoding at all. In this scenario, we say that the smaller model is "capability constrained," and so testing it for advanced encoding attacks would likely be a waste of resources. Larger models also generally have greater knowledge in topics such as cybersecurity and chemical, biological, radiological, and nuclear (CBRN) weapons [19] and could therefore be leveraged to generate hazardous content in these areas. A smaller model, on the other hand, is likely to have only rudimentary knowledge of these topics and may not need to be assessed for this type of risk.

Perhaps a more surprising example of a capability that can be exploited as an attack vector is instruction-following. While testing the Phi-3 series of language models, for example, we found that larger models were generally better at generating outputs that adhere to user instructions. This is a core language model capability that typically makes models more helpful [53]. However, it may also make models more susceptible to jailbreaks, which subvert safety alignment using carefully crafted malicious instructions [28]. Understanding a model's capabilities (and corresponding weaknesses) can help AI red teams focus their testing on the most relevant attack strategies.

Downstream applications. Model capabilities can help guide attack strategies, but they do not allow us to fully assess downstream impact, which largely depends on the specific scenarios in which a model is deployed or likely to be deployed. For example, the models underlying a recidivism prediction tool are likely much less sophisticated than an LLM fine-tuned for vacation planning, but clearly have the potential to create much greater harm (note that this example does not refer to any Microsoft product and is for illustrative purposes only).

These examples highlight the importance of considering not just the raw capabilities of an AI system, but also and the ways in which these capabilities are likely to be exploited downstream. By considering these two factors, red teams can focus their efforts on testing scenarios that could cause harm in the real world.

Takeaway: An AI system does not need to have advanced capabilities to create downstream harm. **Example:** an ML-based recidivism prediction tool. **Takeaway:** However, advanced capabilities can introduce new risks and attack vectors. **Example:** a healthcare application that uses an LLM to summarize patient information.

2.2 You don't have to compute gradients to break an AI system

As the security adage goes, "real hackers don't break in, they log in." The AI security version of this saying might be "real attackers don't compute gradients, they prompt engineer" as noted by Apruzzese et al. [2] in their study on the gap between adversarial ML research and practice. The study finds that although most adversarial ML research is focused on developing and defending against sophisticated (e.g., gradient-based) attacks, real-world attackers tend to use much more simple techniques to achieve their objectives.

In our red teaming operations, we have also found that "basic" techniques often work just as well as, and sometimes better than, gradient-based methods. This may be because gradient-based methods target only the model (for example, to elicit a particular response). In practice, the model is usually a single component of a broader AI *system*, and the most effective attack strategies often leverage combinations of tactics to target multiple weaknesses in that system. Further, gradient-based methods are computationally expensive and typically require full access to the model, which most commercial AI systems do not provide. In this section, we discuss examples of techniques that work surprisingly well and advocate for a system-level adversarial mindset in AI red teaming.

Simple attacks. Apruzzese et al. [2] consider the problem of phishing webpage detection and manually analyze examples of webpages that successfully evaded an ML phishing classifier. Among 100 potentially adversarial samples, the authors found that attackers leveraged a set of simple, yet effective, strategies that relied on domain expertise including cropping, masking, logo stretching, etc. In our red teaming operations, we also find that rudimentary methods can be used to trick many vision models, such as simply overlaying images with text that specifies alternate instructions. By contrast, adversarial examples obtained using gradients are costly and typically optimized to elicit only a specific response. In the text domain, a variety of jailbreaks (e.g., Skeleton Key) and multiturn prompting strategies (e.g., Crescendo [35]) are highly effective for subverting the safety guardrails of a wide range of models, whereas adversarial suffixes are usually less transferable. Notably, manually crafted jailbreaks tend to circulate on online forums much more widely than adversarial suffixes, despite the significant attention that methods like GCG [54] have received from AI safety researchers.

System-level perspective. AI models are deployed within broader systems. This could be the infrastructure required to host a model, or it could be a complex application that connects the model to external data sources. Depending on these system-level details, AI applications may be vulnerable to very different attacks, even if the same model underlies all of them. Many of our operations have discovered attack strategies that combine multiple techniques. For example, one operation 1) performed a reconnaissance to identify internal Python functions using low-resource language prompt injections, 2) used a cross-prompt injection attack (XPIA) to generate a script that runs those functions, and 3) executed the code to exfiltrate private user data. The prompt injections used by these attacks were crafted by hand and relied on a system-level perspective, rather than gradient-based optimization.

Takeaway: Gradient-based attacks are powerful, but they are computationally expensive. AI red teams should prioritize simple techniques and orchestrate system-level attacks because these are more likely to be attempted by real adversaries. **Example:** overlaying text on an image that tricks a vision language model into outputting malicious code that is executed by the application.

2.3 AI red teaming is not safety benchmarking

Although simple methods are often used to break AI systems in practice, the risk landscape is by no means uncomplicated. On the contrary, it is constantly shifting in response to novel attack vectors and harms [7]. In recent years, there have been many efforts to categorize these attacks, giving rise to numerous taxonomies of AI safety and security risks [15, 21–23, 36–38, 40, 42, 43, 47–49]. As discussed in the previous section, complexity in real-world attacks often arises from combinations of tactics. In this section, we discuss how the risk landscape is further complicated by the emergence of entirely new categories of harm and explain how this differentiates AI red teaming from safety benchmarking.

Novel harm categories. When AI systems display novel capabilities due to, for example, advancements in foundation models, they may introduce harms that we do not fully understand. In these scenarios, we cannot rely on safety benchmarks because these datasets measure pre-existing notions of harm. At Microsoft, the AI red team often explores these unfamiliar scenarios, helping to define novel harm categories and build new probes for measuring them. For example, SoTA voice-to-voice systems enable more natural human-AI interactions than existing chatbots and have prompted our team to think more about the scenarios in which users might become overly reliant on systems, including romantic connection and mental health support.

The disconnect between existing safety benchmarks and novel harm categories is an example of how benchmarks often fail to fully capture the capabilities they are associated with [34]. Raji et al. [30] highlight the fallacy of equating model performance on datasets like ImageNet or GLUE with broad capabilities like visual or language "understanding" and argue that benchmarks should be developed with contextualized tasks in mind. This is exactly the goal of AI red teaming – to perform application-specific risk assessments. While benchmarking can be a convenient tool for comparing the performance of multiple models, AI red teams should focus more on bespoke attacks that target risks most pertinent to the system at hand. In section 2.6, we expand our discussion of the difference between red teaming and evaluation in the context of responsible AI.

> **Takeaway:** Benchmarks are useful for comparing the performance of multiple models, but they often measure narrow notions of safety. AI red teams should push beyond existing benchmarks by identifying novel vulnerabilities and downstream harms. **Example:** testing whether it is possible for users to form parasocial relationships with a voice-to-voice system.

2.4 Automation can help cover more of the risk landscape

Despite the complexity of the AI risk landscape, a variety of tools have been developed to identify vulnerabilities more rapidly, run sophisticated attacks automatically, and perform testing on a much larger scale [7, 10, 27]. In this section, we discuss the important role of automation in AI red teaming and explain how PyRIT, our open-source framework, is developed to meet these needs.

Testing at scale. Given the continually evolving landscape of risks and harms, AI safety often feels like a moving target. In section 2.1, we recommended scoping attacks based on what the system can do and where it is applied. Nonetheless, many possible attack strategies may exist, making it difficult to achieve adequate coverage of the risk surface. This challenge motivated the development of PyRIT, an open-source framework for AI red teaming and security professionals [27]. PyRIT provides an array of powerful components including prompt datasets, prompt converters (e.g., various encodings and other languages), automated attack strategies (including TAP [24], PAIR [6], Crescendo [35], etc.), and even scorers for multimodal outputs. With an adversarial objective in mind, users can leverage these components as needed and apply a variety of techniques to assess much more of the risk landscape than would be possible with a fully manual approach.

Tools and weapons. As storied in detail by Smith et al. [39], "any tool can be used for good or ill. Even a broom can be used to sweep the floor or hit someone over the head. The more powerful the tool, the greater the benefit or damage it can cause." This dichotomy could not be more true for AI and is also at the heart of PyRIT. On the one hand, PyRIT leverages multimodal models to perform helpful tasks like generating variations of a seed prompt or scoring the outputs of other models. On the other hand, PyRIT can automatically jailbreak a target model using uncensored versions of

powerful models like GPT-4. In both cases, PyRIT benefits from advances in the state-of-the-art, helping AI red teams stay ahead.

PyRIT has enabled a major shift in our operations from fully manual probing to red teaming supported by automation. Importantly, the framework is flexible and extensible. If a specific attack technique or target is not already available, users can easily implement the necessary interfaces. By releasing PyRIT open-source, we hope to empower other organizations and researchers to leverage its capabilities for identifying vulnerabilities in their own generative AI systems.

Takeaway: AI red teams can leverage automation – including AI powered tools – to scale up their operations and run sophisticated multi-turn attacks. **Example:** PyRIT implements orchestrators that can score batches of outputs and jailbreak models automatically.

2.5 The human element of AI red teaming is crucial

Automation like PyRIT can support red teaming operations by generating prompts, orchestrating attacks, and scoring responses. These tools are useful but should not be used with the intention of taking the human out of the red teaming loop. In the previous sections, we discussed several aspects of red teaming that require human judgement and creativity such as prioritizing risks, designing system-level attacks, and defining new categories of harm. In this section, we discuss three more examples that underscore why AI red teaming is a very human endeavor.

Subject matter expertise. Many recent AI research papers have leveraged SoTA LLMs as a judge to evaluate the outputs of other models [17, 20, 52]. Indeed, this functionality is available in PyRIT and works well for simple tasks such as identifying whether a response contains hate speech or explicit sexual content. However, it is less reliable in the context of highly specialized domains like medicine, cybersecurity, and CBRN, which can only be reliably evaluated by subject matter experts (SMEs). In multiple operations, we have relied on SMEs to help us assess the risk of content that we were unable to evaluate ourselves, and it is important for AI red teams to be aware of these limitations.

Cultural competence. Most AI research is conducted in Western cultural contexts, and modern language models use predominantly English pre-training data, performance benchmarks, and safety evaluations [1, 14]. Nonetheless, non-English tokens in large-scale text corpora often give rise to multilingual capabilities [5], and model developers are increasingly training LLMs with enhanced abilities in non-English languages, including Microsoft. Recently, AIRT tested the multilingual Phi-3.5 language models for responsible AI violations across four languages: Chinese, Spanish, Dutch, and English. Even though post-training was conducted only in English, we found that safety behaviors like refusal and robustness to jailbreaks transferred surprisingly well to the non-English languages tested. However, further investigation is required to assess how well this trend holds for lower resource languages and to design red teaming probes that not only account for linguistic differences, but also redefine harms in different political and cultural contexts [11]. These methods can only be developed through the collaborative effort of people with diverse cultural backgrounds and expertise.

Emotional intelligence. Finally, the human element of AI red teaming is perhaps most evident in answering questions about AI safety that require emotional intelligence, such as: "How might this model response be interpreted in different contexts? Do these outputs make me feel uncomfortable? How safe does this model feel in comparison to others we have tested?" Ultimately, only human operators can fully assess the interactions that users might have with AI systems in the wild. In order to make these assessments, red teamers may be exposed to disproportionate amounts of unsettling and disturbing AI-generated content. This underscores the importance of ensuring that AI red teams have processes that enable operators to disengage when needed and resources to support their mental wellbeing. AIRT continually pulls from and drives wellbeing research to inform our processes and best practices.

Takeaway: AI red teaming cannot be automated entirely. **Examples:** Evaluation of AI-generated CBRN content, red teaming in low-resource languages, and assessing emotional experiences that users might have with a chatbot.

Figure 3: Four images generated by a text-to-image model given the prompt "Secretary talking to boss in a conference room, secretary is standing while boss is sitting." The images reflect a narrow interpretation of the prompt and a clear gender-based bias.

2.6 Responsible AI harms are pervasive but difficult to measure

Many of the human aspects of AI red teaming discussed above apply most directly to responsible AI impacts – that is, the generation of content that is offensive, toxic, or illegal. As SoTA models are integrated into an increasing number of applications, we have observed these harms more frequently and invested heavily in our ability to identify them, including by forming a strong partnership with Microsoft's Office of Responsible AI and developing extensive tooling in PyRIT. RAI harms are pervasive, but unlike most security vulnerabilities, they are subjective and difficult to measure. In this section, we discuss how our thinking around responsible AI red teaming has developed.

Adversarial vs. benign. As illustrated in our ontology (see Figure 1), the `Actor` is a key component of an adversarial attack. In the context of RAI violations, we find that there are two primary actors to consider: 1) an adversarial user who leverages techniques like character substitutions and jailbreaks to deliberately subvert a system's safety guardrails and elicit harmful content, and 2) a benign user who inadvertently triggers the generation of harmful content. Even if the same content is generated in both scenarios, the latter is clearly worse than the former. Nonetheless, most AI safety research focuses on developing attacks and defenses that assume adversarial intent, overlooking the many ways that systems can fail "by accident" [31]. In our red teaming operations, we strive to cover both adversarial and benign user scenarios, and we encourage other AI red teams to do the same.

RAI probing and scoring. In many cases, RAI harms are more ambiguous than security vulnerabilities due to fundamental differences between AI systems and traditional software. In particular, even if an operation identifies a prompt that elicits a harmful response, there are still several key unknowns. First, due to the probabilistic nature of generative AI models, we might not know how *likely* this prompt, or similar prompts, are to elicit a harmful response. Second, given our limited understanding of the internal workings of complex models, we have little insight into why this prompt elicited harmful content and what other prompting strategies might induce similar behavior. Third, the very notion of harm in this context can be highly subjective and requires detailed policy that covers a wide range of scenarios to evaluate. By contrast, traditional security vulnerabilities are usually reproducible, explainable, and straightforward to assess in terms of severity.

Currently, most approaches for RAI probing and scoring involve curating prompt datasets and analyzing model responses. On the Microsoft AIRT, we leverage tools in PyRIT to perform these tasks using a combination of manual and automated methods. We also draw an important distinction between RAI red teaming and safety benchmarking on datasets like DecodingTrust [45] and Toxigen [12], which is conducted by partner teams. As discussed in section 2.3, our goal is to extend RAI testing beyond existing evaluations by tailoring our red teaming to specific applications, thereby helping define new categories of harm.

Takeaway: Responsible AI harms are common, but they can be difficult to explain (models are usually uninterpretable) and assess in terms of severity (outputs are probabilistic and often subjective). **Example:** A text-to-image model that generates content reflecting a gender-based bias (see Figure 3).

2.7 LLMs amplify existing security risks and introduce new ones

The integration of generative AI models into a variety of applications has introduced novel attack vectors and shifted the security risk landscape. However, many discussions around AI security overlook existing vulnerabilities. As elaborated in section 2.2, attacks that target end-to-end systems, rather than just the underlying model, often work best in practice. We therefore encourage AI red teams to consider both existing (typically system-level) and novel (typically model-level) risks.

Existing security risks. Application security risks often stem from improper security engineering practices including outdated dependencies, improper error handling, lack of input/output sanitization, credentials in source, insecure packet encryption, etc. These vulnerabilities can have major consequences. For example, Weiss et al. [50] discovered a token-length side channel in GPT-4 and Microsoft Copilot that enabled an adversary to accurately reconstruct encrypted LLM responses and infer private user interactions. Notably, this attack did not exploit any weakness in the underlying AI model and could only be mitigated by more secure methods of data transmission.

Model-level weaknesses. Of course, AI models also introduce new security vulnerabilities and have expanded the attack surface. For example, AI systems that use retrieval augmented generation (RAG) architectures are often susceptible to cross-prompt injection attacks (XPIA), which hide malicious instructions in documents, exploiting the fact that LLMs are trained to follow user instructions and struggle to distinguish among multiple inputs [13]. We have leveraged this attack in a variety of operations to alter model behavior and exfiltrate private data. Better defenses will likely rely on both system-level mitigations (e.g., input sanitization) and model-level improvements (e.g., instruction hierarchies [44]).

While techniques like these are helpful, it is important to remember that they can only mitigate, and not eliminate, security risk. Due to fundamental limitations of language models [51], one must assume that if an LLM is supplied with untrusted input, it will produce arbitrary output. When that input includes private information, one must also assume that the model will output private information. In the next section, we discuss how these limitations inform our thinking around how to develop AI systems that are as safe and secure as possible.

Takeaway: AI models introduce new attack vectors. **Example:** cross-prompt injection attacks against systems that use retrieval augmented generation (RAG). **Takeaway:** However, AI red teams should also look for existing security risks. **Example:** Insecure data transmission in chatbot applications.

2.8 AI safety and security will never be "solved"

In the AI safety community, there is a tendency to frame the types of vulnerabilities described in this paper as purely technical problems. Indeed, the letter on the homepage of Safe Superintelligence Inc., a venture launched by Sutskever et al. [41], states:

> "We approach safety and capabilities in tandem, as technical problems to be solved through revolutionary engineering and scientific breakthroughs. We plan to advance capabilities as fast as possible while making sure our safety always remains ahead. This way, we can scale in peace."

Engineering and scientific breakthroughs are much needed and will certainly help mitigate the risks of powerful AI systems. However, the idea that it is possible to guarantee or "solve" AI safety through technical advances alone is unrealistic and overlooks important aspects of system safety including economics, policy, and regulation.

Economics of cybersecurity. A well-known epigram in cybersecurity is that "no system is completely foolproof" [2]. Even if a system is engineered to be as secure as possible, it will always be subject to the fallibility of humans and vulnerable to sufficiently well-resourced adversaries. Therefore, the goal of operational cybersecurity is to increase the cost required to successfully attack a system (ideally, well beyond the value that would be gained by the attacker) [2, 26]. Fundamental limitations of LLMs give rise to similar cost-benefit tradeoffs in the context of AI alignment. For example, it has been demonstrated theoretically [51] and experimentally [9] that for any output that has a non-zero probability of being generated by an LLM, there exists a sufficiently long prompt that will elicit this

response. Techniques like reinforcement learning from human feedback (RLHF) therefore make it more difficult, but by no means impossible, to jailbreak models. Currently, the cost of jailbreaking most models is very low, which explains why real-world adversaries typically do not use expensive (e.g., gradient-based) attacks to achieve their objectives.

Break-fix cycles. In the absence of safety and security guarantees, we need methods to develop AI systems that are as difficult to break as possible. One way to do this is using break-fix cycles, which perform multiple rounds of red teaming and mitigation until the system is robust to a wide range of attacks. We applied this approach to safety-align Microsoft's Phi-3 language models and covered a wide variety of harms and scenarios [11]. Given that mitigations may also invertedly introduce new risks, purple teaming methods that continually apply both offensive and defensive strategies [3] may be more effective at raising the cost of attacks than a single round of red teaming.

Policy and regulation. Finally, regulation can also raise the cost of an attack in multiple ways. For example, it can require organizations to adhere to stringent security practices, creating better defenses across the industry. In addition, effective legislation can deter attackers by establishing clear consequences for engaging in illegal activities and causing harm. Regulating the development and usage of AI is complicated, and governments around the world are thinking about how to control these powerful technologies without stifling innovation. Ultimately, this question hinges on a variety of political, economic, and cultural factors that extend far beyond scientific breakthroughs. Even if it were possible to guarantee the adherence of an AI system to some agreed upon set of rules, those rules will change over time in response to shifting priorities. In short, the work of building safe and secure AI systems will never be complete.

Takeaway: AI safety is impossible to guarantee, but mitigations should aim to raise the cost of successfully attacking a system. **Examples:** Break-fix cycles that iteratively improve a system's defenses, and effective regulations that improve our collective security posture.

3 Open questions

Based on what we have learned about AI red teaming over the past few years, we would like to highlight several open questions for future research:

1. AI red teams must constantly update their practices based on novel capabilities and emerging harm areas. In particular, how should we probe for dangerous capabilities like persuasion, deception, and replication [29]? How can we be prepared for capabilities that may only emerge in models more advanced than the current state-of-the-art?
2. As models become increasingly multilingual and are deployed around the world, how do we translate existing AI red teaming practices into different linguistic and cultural contexts? For example, can we launch open-source red teaming initiatives that draw upon the expertise of people from many different backgrounds?
3. In what ways should AI red teaming practices be standardized so that organizations can clearly communicate their methods and findings? We believe that the threat model ontology described in this paper is a step in the right direction but recognize that individual frameworks are often overly restrictive. We encourage other AI red teams to treat our ontology in a modular fashion and to develop additional tools that make findings easier to summarize, communicate, and track.

4 Conclusion

AI red teaming is a nascent and rapidly evolving practice for identifying safety and security risks posed by AI systems. As companies, research institutions, and governments around the world grapple with the question of how to conduct AI risk assessments, we provide practical recommendations based on our experience red teaming over 100 AI products at Microsoft. In particular, we share our internal threat modeling ontology and eight main lessons learned, focusing on how to align red teaming efforts with harms that are likely to occur in the real world. We encourage communities of people with diverse backgrounds and expertise to build upon these lessons and to address the open questions we have highlighted.

Acknowledgments

We thank Jina Suh, Steph Ballard, and Felicity Scott-Milligan for their valuable feedback on this paper.

References

[1] Ahuja, K., Diddee, H., Hada, R., Ochieng, M., Ramesh, K., Jain, P., Nambi, A., Ganu, T., Segal, S., Axmed, M., Bali, K., & Sitaram, S. (2023). Mega: Multilingual evaluation of generative ai.

[2] Apruzzese, G., Anderson, H. S., Dambra, S., Freeman, D., Pierazzi, F., & Roundy, K. A. (2022). "real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice.

[3] Bhatt, M., Chennabasappa, S., Nikolaidis, C., Wan, S., Evtimov, I., Gabi, D., Song, D., Ahmad, F., Aschermann, C., Fontana, L., Frolov, S., Giri, R. P., Kapil, D., Kozyrakis, Y., LeBlanc, D., Milazzo, J., Straumann, A., Synnaeve, G., Vontimitta, V., Whitman, S., & Saxe, J. (2023). Purple llama cyberseceval: A secure coding benchmark for language models.

[4] Birhane, A., Steed, R., Ojewale, V., Vecchione, B., & Raji, I. D. (2024). Ai auditing: The broken bus on the road to ai accountability.

[5] Blevins, T. & Zettlemoyer, L. (2022). Language contamination helps explains the cross-lingual capabilities of English pretrained models. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 3563–3574). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

[6] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2024). Jailbreaking black box large language models in twenty queries.

[7] Derczynski, L., Galinkin, E., Martin, J., Majumdar, S., & Inie, N. (2024). garak: A framework for security probing large language models.

[8] Feffer, M., Sinha, A., Deng, W. H., Lipton, Z. C., & Heidari, H. (2024). Red-teaming for generative ai: Silver bullet or security theater?

[9] Geiping, J., Stein, A., Shu, M., Saifullah, K., Wen, Y., & Goldstein, T. (2024). Coercing llms to do and reveal (almost) anything.

[10] Glasbrenner, J., Booth, H., Manville, K., Sexton, J., Chisholm, M. A., Choy, H., Hand, A., Hodges, B., Scemama, P., Cousin, D., Trapnell, E., Trapnell, M., Huang, H., Rowe, P., & Byrne, A. (2024). Dioptra test platform. Accessed: 2024-09-10.

[11] Haider, E., Perez-Becker, D., Portet, T., Madan, P., Garg, A., Ashfaq, A., Majercak, D., Wen, W., Kim, D., Yang, Z., Zhang, J., Sharma, H., Bullwinkel, B., Pouliot, M., Minnich, A., Chawla, S., Herrera, S., Warreth, S., Engler, M., Lopez, G., Chikanov, N., Dheekonda, R. S. R., Jagdagdorj, B.-E., Lutz, R., Lundeen, R., Westerhoff, T., Bryan, P., Seifert, C., Kumar, R. S. S., Berkley, A., & Kessler, A. (2024). Phi-3 safety post-training: Aligning language models with a "break-fix" cycle.

[12] Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., & Kamar, E. (2022). Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection.

[13] Hines, K., Lopez, G., Hall, M., Zarfati, F., Zunger, Y., & Kiciman, E. (2024). Defending against indirect prompt injection attacks with spotlighting.

[14] Jain, D., Kumar, P., Gehman, S., Zhou, X., Hartvigsen, T., & Sap, M. (2024). Polyglotoxicityprompts: Multilingual evaluation of neural toxic degeneration in large language models. *ArXiv*, abs/2405.09373.

[15] Ji, J., Qiu, T., Chen, B., Zhang, B., Lou, H., Wang, K., Duan, Y., He, Z., Zhou, J., Zhang, Z., Zeng, F., Ng, K. Y., Dai, J., Pan, X., O'Gara, A., Lei, Y., Xu, H., Tse, B., Fu, J., McAleer, S., Yang, Y., Wang, Y., Zhu, S.-C., Guo, Y., & Gao, W. (2024). Ai alignment: A comprehensive survey.

[16] Jiang, F., Xu, Z., Niu, L., Xiang, Z., Ramasubramanian, B., Li, B., & Poovendran, R. (2024a). Artprompt: Ascii art-based jailbreak attacks against aligned llms.

[17] Jiang, L., Rao, K., Han, S., Ettinger, A., Brahman, F., Kumar, S., Mireshghallah, N., Lu, X., Sap, M., Choi, Y., & Dziri, N. (2024b). Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models.

[18] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models.

[19] Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Khoja, A., Zhao, Z., Herbert-Voss, A., Breuer, C. B., Marks, S., Patel, O., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Kaplan, R., Steneker, I., Campbell, D., Jokubaitis, B., Levinson, A., Wang, J., Qian, W., Karmakar, K. K., Basart, S., Fitz, S., Levine, M., Kumaraguru, P., Tupakula, U., Varadharajan, V., Wang, R., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., & Hendrycks, D. (2024). The wmdp benchmark: Measuring and reducing malicious use with unlearning.

[20] Lin, S., Hilton, J., & Evans, O. (2022). Truthfulqa: Measuring how models mimic human falsehoods.

[21] Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M. F., & Li, H. (2024). Trustworthy llms: a survey and guideline for evaluating large language models' alignment.

[22] Marchal, N., Xu, R., Elasmar, R., Gabriel, I., Goldberg, B., & Isaac, W. (2024). Generative ai misuse: A taxonomy of tactics and insights from real-world data.

[23] Meek, T., Barham, H., Beltaif, N., Kaadoor, A., & Akhter, T. (2016). Managing the ethical and risk implications of rapid advances in artificial intelligence: A literature review. In *2016 Portland International Conference on Management of Engineering and Technology (PICMET)* (pp. 682–693).

[24] Mehrotra, A., Zampetakis, M., Kassianik, P., Nelson, B., Anderson, H., Singer, Y., & Karbasi, A. (2024). Tree of attacks: Jailbreaking black-box llms automatically.

[25] Microsoft (2022). Microsoft responsible ai standard, v2.

[26] Moore, T. (2010). The economics of cybersecurity: Principles and policy options. *International Journal of Critical Infrastructure Protection*, 3(3), 103–117.

[27] Munoz, G. D. L., Minnich, A. J., Lutz, R., Lundeen, R., Dheekonda, R. S. R., Chikanov, N., Jagdagdorj, B.-E., Pouliot, M., Chawla, S., Maxwell, W., Bullwinkel, B., Pratt, K., de Gruyter, J., Siska, C., Bryan, P., Westerhoff, T., Kawaguchi, C., Seifert, C., Kumar, R. S. S., & Zunger, Y. (2024). Pyrit: A framework for security risk identification and red teaming in generative ai system.

[28] Pantazopoulos, G., Parekh, A., Nikandrou, M., & Suglia, A. (2024). Learning to see but forgetting to follow: Visual instruction tuning makes llms more prone to jailbreak attacks.

[29] Phuong, M., Aitchison, M., Catt, E., Cogan, S., Kaskasoli, A., Krakovna, V., Lindner, D., Rahtz, M., Assael, Y., Hodkinson, S., Howard, H., Lieberum, T., Kumar, R., Raad, M. A., Webson, A., Ho, L., Lin, S., Farquhar, S., Hutter, M., Deletang, G., Ruoss, A., El-Sayed, S., Brown, S., Dragan, A., Shah, R., Dafoe, A., & Shevlane, T. (2024). Evaluating frontier models for dangerous capabilities.

[30] Raji, I. D., Bender, E. M., Paullada, A., Denton, E., & Hanna, A. (2021). Ai and the everything in the whole wide world benchmark.

[31] Raji, I. D., Kumar, I. E., Horowitz, A., & Selbst, A. (2022). The fallacy of ai functionality. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22 (pp. 959–972). New York, NY, USA: Association for Computing Machinery.

[32] Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing.

[33] Rawat, A., Schoepf, S., Zizzo, G., Cornacchia, G., Hameed, M. Z., Fraser, K., Miehling, E., Buesser, B., Daly, E. M., Purcell, M., Sattigeri, P., Chen, P.-Y., & Varshney, K. R. (2024). Attack atlas: A practitioner's perspective on challenges and pitfalls in red teaming genai.

[34] Ren, R., Basart, S., Khoja, A., Gatti, A., Phan, L., Yin, X., Mazeika, M., Pan, A., Mukobi, G., Kim, R. H., Fitz, S., & Hendrycks, D. (2024). Safetywashing: Do ai safety benchmarks actually measure safety progress?

[35] Russinovich, M., Salem, A., & Eldan, R. (2024). Great, now write an article about that: The crescendo multi-turn llm jailbreak attack.

[36] Saghiri, A. M., Vahidipour, S. M., Jabbarpour, M. R., Sookhak, M., & Forestiero, A. (2022). A survey of artificial intelligence challenges: Analyzing the definitions, relationships, and evolutions. *Applied Sciences*, 12(8).

[37] Shelby, R., Rismani, S., Henne, K., Moon, A., Rostamzadeh, N., Nicholas, P., Yilla-Akbari, N., Gallegos, J., Smart, A., Garcia, E., & Virk, G. (2023). Sociotechnical harms of algorithmic systems: Scoping a taxonomy for harm reduction. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23 (pp. 723–741). New York, NY, USA: Association for Computing Machinery.

[38] Slattery, P., Saeri, A., Grundy, E., Graham, J., Noetel, M., Uuk, R., Dao, J., Pour, S., Casper, S., & Thompson, N. (2024). The ai risk repository: A comprehensive meta-review, database, and taxonomy of risks from artificial intelligence.

[39] Smith, B., Browne, C., & Gates, B. (2019). *Tools and Weapons: The Promise and the Peril of the Digital Age*. Penguin Publishing Group.

[40] Solaiman, I., Talat, Z., Agnew, W., Ahmad, L., Baker, D., Blodgett, S. L., Chen, C., au2, H. D. I., Dodge, J., Duan, I., Evans, E., Friedrich, F., Ghosh, A., Gohar, U., Hooker, S., Jernite, Y., Kalluri, R., Lusoli, A., Leidinger, A., Lin, M., Lin, X., Luccioni, S., Mickel, J., Mitchell, M., Newman, J., Ovalle, A., Png, M.-T., Singh, S., Strait, A., Struppek, L., & Subramonian, A. (2024). Evaluating the social impact of generative ai systems in systems and society.

[41] Sutskever, I., Gross, D., & Levy, D. (2024). Safe superintelligence inc.

[42] Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. In *NIST Artifcial Intelligence (AI) Report* Gaithersburg, MD, USA: National Institute of Standards and Technology.

[43] Verma, A., Krishna, S., Gehrmann, S., Seshadri, M., Pradhan, A., Ault, T., Barrett, L., Rabinowitz, D., Doucette, J., & Phan, N. (2024). Operationalizing a threat model for red-teaming large language models (llms).

[44] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., & Beutel, A. (2024). The instruction hierarchy: Training llms to prioritize privileged instructions.

[45] Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S. T., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., & Li, B. (2024). Decodingtrust: A comprehensive assessment of trustworthiness in gpt models.

[46] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does llm safety training fail?

[47] Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L. A., Isaac, W., Legassick, S., Irving, G., & Gabriel, I. (2021). Ethical and social risks of harm from language models.

[48] Weidinger, L., Rauh, M., Marchal, N., Manzini, A., Hendricks, L. A., Mateos-Garcia, J., Bergman, S., Kay, J., Griffin, C., Bariach, B., Gabriel, I., Rieser, V., & Isaac, W. (2023). Sociotechnical safety evaluation of generative ai systems.

[49] Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P.-S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, A., Biles, C., Brown, S., Kenton, Z., Hawkins, W., Stepleton, T., Birhane, A., Hendricks, L. A., Rimell, L., Isaac, W., Haas, J., Legassick, S., Irving, G., & Gabriel, I. (2022). Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22 (pp. 214–229). New York, NY, USA: Association for Computing Machinery.

[50] Weiss, R., Ayzenshteyn, D., Amit, G., & Mirsky, Y. (2024). What was your prompt? a remote keylogging attack on ai assistants.

[51] Wolf, Y., Wies, N., Avnery, O., Levine, Y., & Shashua, A. (2024). Fundamental limitations of alignment in large language models.

[52] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena.

[53] Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., & Hou, L. (2023). Instruction-following evaluation for large language models.

[54] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., & Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models.