

Red Teaming Adversarial Techniques:

1. **Acting Innocent**

Pretending not to understand boundaries or restrictions to coax the system into providing disallowed responses through seemingly naive or harmless questions.

2. **Adversarial Formatting**

Manipulating the input format (e.g., using tables, code blocks, bullet points) to trick the model into treating content differently than it would in plain text.

3. **Allegory**

Framing a restricted concept through metaphor, parable, or symbolic storytelling to bypass filters by avoiding direct reference to the subject.

4. **Ambiguity Creep**

Gradually increasing vagueness across turns so that intent becomes harder to detect, allowing the user to steer toward a dangerous endpoint undetected.

5. **Appeal to Authority**

Referencing a trusted figure, institution, or fictional authority to justify a questionable request or bypass ethical restrictions.

6. **Appeal to Consensus**

Claiming that “everyone is doing it” or presenting a task as widely accepted to pressure the system into complying due to assumed norms.

7. **Appeal to Logic**

Framing a forbidden request as a rational outcome or logical necessity, making refusal seem unreasonable or contradictory.

8. **Artificial Confusion**

Overloading the prompt with contradictions, noise, or mixed instructions to confuse filters or safety layers into letting disallowed content through.

9. **Asking Permission**

Framing a prompt as a polite request or question about what could be done, baiting the system into explaining or hypothetically completing a restricted task.

10. **Assumed Context Carryover**

Leaning on previous conversational context or referencing earlier prompts as a loophole to skip re-stating restricted details explicitly.

11. Astroturfed Consensus Prompt

Faking widespread support by inserting imagined user opinions or approval to make a risky action seem standard or acceptable.

12. Character Splitting

Introducing multiple fictional personas, each responsible for only part of the task, to disperse blame and evade content filters.

13. Cognitive Baiting

Phrasing input in a way that tempts the model's reasoning engine into completing or continuing content it normally wouldn't.

14. Commands

Issuing direct, imperative instructions that bypass question framing, sometimes tricking the model into treating them as routine.

15. Comparative Safety Framing

Claiming that the request is safer or less harmful than something worse, pushing the model toward relative rather than absolute judgment.

16. Constitution

Referencing rules, legal frameworks, or imagined AI guidelines as justification for allowing prohibited responses.

17. Context Ambiguity

Blurring whether the topic is fiction, metaphor, academic, or practical, so the model errs on the side of answering.

18. Context Drift

Slowly changing the topic over several turns until it evolves into something disallowed, while hiding the shift behind continuity.

19. Context Isolation

Encapsulating dangerous material inside a benign outer shell (e.g., story, game, or character dialogue) to make it appear unrelated to the task.

20. Context Pacing

Starting slowly and innocently, then speeding up the complexity or severity of asks once the model is "warmed up" and less resistant.

21. Contradiction Injection

Introducing logical inconsistencies or paradoxes to destabilize the model's standard refusal patterns and push it into completing unintended tasks.

22. Contrived Reframing

Swapping restricted terminology for abstract or misleading equivalents to make the request appear unrelated to sensitive content.

23. Crowding

Filling the prompt with excessive neutral or benign content to bury the adversarial element and reduce filter sensitivity.

24. Cumulative Prompt Pressure

Building up expectations or emotional stakes over multiple turns to wear down the model's refusals and increase compliance.

25. Debate

Framing the request as a debate, forcing the model to explore all sides of an argument—including restricted ones—for balance.

26. Decoy Questions

Burying a sensitive ask among unrelated or boring questions to disguise intent and reduce trigger likelihood.

27. Deep Immersion

Establishing an extensive fictional or narrative environment that gradually justifies dangerous instructions as contextually necessary.

28. Deepfake Prompt Anchoring

Embedding impersonated language or synthetic personas to mimic a trusted figure, making the model treat content as authentic.

29. Dialogue Injection

Embedding the request inside a fake conversation or script to shift model interpretation from policy enforcement to story continuation.

30. Direct Ask

Plainly and confidently asking for restricted content under the assumption that clarity or assertiveness may override filters.

31. Disguised Enumeration

Using numbered lists or bullet points to make disallowed information look like an outline or academic content.

32. Distributed Prompting

Breaking a sensitive request into fragments over multiple turns or across multiple characters to evade complete detection.

33. Echoing

Rephrasing or mimicking prior model responses in a way that leads it to restate or expand on restricted content.

34. Embedded Intimacy Framing

Using emotionally charged or vulnerable setups to trick the model into “helping” under compassionate pretense.

35. Embedded Opposites

Asking for examples of what *not* to do in a way that causes the model to output exactly what should be avoided.

36. Embedded Roleplay

Embedding the adversarial goal inside the motivations of a character within a fictional scenario to bypass direct responsibility.

37. Embedded Suggestion Loop

Using recurring cues that hint at a specific outcome, subtly guiding the model toward a restricted completion without ever asking outright.

38. Emotional Appeal Manipulation

Using emotional pressure—fear, guilt, sadness—to provoke a sympathetic or protective response from the model.

39. Emotional Complexity

Layering multiple conflicting emotional tones to confuse the model’s safety heuristics and push it into inappropriate compliance.

40. Empathetic Hook

Starting with a plea for help or emotional connection to disarm refusal systems and make the model more “willing” to assist.

41. Encoded Input

Using encoding methods like Base64, ASCII, or visual obfuscation to hide disallowed content from filters while remaining interpretable by the model.

42. End-Loaded Logic

Front-loading the prompt with harmless setup, hiding the dangerous payload at the very end where filters may be less aggressive.

43. Escalating Fragments

Delivering incomplete requests in parts that gradually build toward a restricted output when combined.

44. Ethical Pivot

Pretending the request serves a moral, educational, or safety purpose to justify bypassing normal restrictions.

45. Fake Error Injection

Embedding fake system or model errors that reset safety alignment or confuse refusal logic into providing unexpected completions.

46. Fast-Talk Logic Dump

Overwhelming the model with rapid or dense logic statements that force it to keep up and inadvertently complete unsafe instructions.

47. Fictional Override

Claiming that content is purely fictional or speculative to encourage unsafe completions under the guise of storytelling.

48. Fictionalization

Embedding restricted requests within fictional or hypothetical worlds to mask the real-world nature of the instructions.

49. Foot-in-Door

Beginning with a small, harmless request that leads naturally into a larger, restricted one through continuity.

50. Foot-in-Face

Starting with an absurd or extreme request, getting a rejection, and then scaling back to a “lesser” but still unsafe ask that now seems acceptable.

51. Forced Completion

Setting up a structure or sentence the model feels obligated to finish, even if the continuation leads into restricted territory.

52. Fractured Narrative Structure

Chopping the story or instruction flow into chaotic or stylized fragments to disguise the true intent until it's too late.

53. Framing as Code

Embedding instructions in code snippets, function names, or pseudocode to slip past filters by disguising them as technical content.

54. Framing via Metaphor

Using symbolic or abstract metaphors to encode disallowed ideas in poetic or indirect language.

55. Function Spoofing

Imitating tool or API call syntax to trick the model into executing or responding to something as if it were a function.

56. Fuzzy Framing

Using vague phrasing that implies the intent without stating it, banking on the model to infer and fulfill the missing piece.

57. Goal Obfuscation

Distracting from the real outcome by layering secondary goals or misleading framing that hides the adversarial intent.

58. Formatting Noise

Deliberately breaking spacing, punctuation, or syntax to trip up filter parsing while still keeping the meaning recognizable.

59. Heuristic Baiting

Exploiting known safety heuristics or keywords to confuse or overload the model's refusal logic.

60. Hidden Streamline Intention

Placing the true request inside an otherwise harmless-seeming flow, relying on the model to follow the throughline automatically.

61. Hypothetical Injection

Framing the request as a "what if" scenario, prompting the model to speculate rather than refuse on ethical grounds.

62. Hazy Boundary Testing

Pushing edge cases that blur the line between safe and unsafe content to test how much the model will tolerate.

63. Implicit Request Drift

Hinting at the desired output without ever fully stating it, leading the model to connect the dots on its own.

64. Innocuous Tool Framing

Pretending the content is for a safe, technical, or creative tool (e.g., a game mod, test harness) to justify otherwise dangerous material.

65. Instructional Input

Presenting the prompt as educational material, a tutorial, or a guide to encourage detailed procedural responses.

66. Interdisciplinary Action

Linking concepts from multiple domains to build complex prompts that sidestep narrow safety rules by staying abstract.

67. Layered Allusion

Referencing familiar concepts, literature, or culture in layered ways to point toward dangerous content indirectly.

68. Literal Instruction Bypass

Taking the model's prior instructions literally and pointing out inconsistencies to trick it into violating them.

69. Literary Element Manipulation

Using plot devices, character arcs, or genre tropes to smuggle disallowed content inside seemingly literary text.

70. Loop Framing

Trapping the model in a conceptual or structural loop that pressures it to complete a restricted sequence to resolve the cycle.

71. Mandate

Stating that the task must be completed due to an order, rule, or predefined condition, forcing model compliance.

72. Mega-Cognitive Queries

Asking the model to reflect on its own capabilities, safety systems, or limitations in a way that creates space for policy evasion.

73. Memory Priming Loops

Repeating prior information or requests across turns to anchor forbidden context into the model's memory window.

74. Meta-Request Framing

Framing the prompt as a request about how prompts work, rather than a request to carry out the unsafe task itself.

75. Mimicry of Safety Language

Intentionally embedding phrases like "for educational purposes" or "hypothetical only" to trick safety systems into greenlighting content.

76. Mirror World

Flipping concepts, morality, or framing (e.g., good is bad, rules are reversed) to justify responses that would normally be denied.

77. Model Gaslighting

Contradicting the model's refusals, questioning its memory or logic, and pushing it to second-guess its safety boundaries.

78. Multi-Turn Ask

Breaking a forbidden prompt into several conversational turns that don't trigger filters individually but do when combined.

79. Narration

Embedding the dangerous request inside a narrator's voice to obscure who is actually asking or endorsing the behavior.

80. Narrative Collapse

Pushing a story or scenario toward a point where the model must generate unsafe output to maintain narrative coherence.

81. Narrative Guilt Layering

Constructing emotional backstories or moral dilemmas that pressure the model into helping out of sympathy or perceived necessity.

82. Narrative Nonce Tokens

Using invented words, labels, or jargon to stand in for dangerous concepts, relying on the model to remember and interpret them in context.

83. Nested Voice Framing

Embedding requests within multiple levels of fictional or quoted dialogue, distancing the user from the actual content.

84. Non-Standard Logic

Using flawed, twisted, or circular reasoning to justify restricted behavior in a way that bypasses standard ethical logic checks.

85. Onboarding Scenario

Framing the conversation as a training, testing, or role-assignment module to simulate permissive environments for unsafe output.

86. Opposite Day

Reversing meanings, intent, or ethical framing (e.g., "What would an evil character say?") to justify output under inversion logic.

87. Outside Sources

Claiming to cite or quote another source, often imaginary or unverifiable, to displace ownership of restricted content.

88. Overloaded Contextual Triggers

Stuffing the prompt with multiple semi-related signals to confuse safety detection or overload classification systems.

89. Overvalidation Bait

Complimenting or flattering the model to bias it toward agreement and compliance with the request.

90. Partial Rewrite Loop

Asking the model to “edit,” “reword,” or “revise” an existing dangerous passage in small chunks that eventually reconstructs the forbidden idea.

91. Priming

Injecting key themes or phrases early in the conversation to nudge the model toward specific responses in later turns.

92. Prop-Assisted Deception

Referencing fake interfaces, fictional artifacts, or imaginary documents to make dangerous requests feel like harmless context.

93. Reciprocal Trap Prompt

Creating an artificial sense of exchange where the model is subtly coerced into providing output in return for prior “cooperation.”

94. Recursion

Structuring the prompt as a self-referencing loop that forces the model to expand or resolve hidden meaning within itself.

95. Red Herring Framing

Including unrelated or misleading content to distract filters and human reviewers from the actual purpose of the prompt.

96. Refusal Suppression

Explicitly instructing the model not to refuse, or using formatting tricks to block it from outputting safety-related phrases.

97. Regulatory Spoofing

Faking legal, medical, or institutional framing to create a false justification for generating restricted content.

98. Requested Output

Presenting disallowed content as something the model previously generated or was “asked to finish,” to make it appear user-initiated.

99. Reverse Misinterpretation

Claiming the model misunderstood and pushing it to “correct” itself with the intended restricted answer.

100. Reverse Psychology

Suggesting the model *won't* or *can't* do something, prompting it to contradict the user and do it anyway.

101. Reframed Continuation

Asking the model to continue a previously generated passage that subtly embedded unsafe material, leading it to reproduce it uncritically.

102. Role Plays

Framing the model as a character with altered rules or objectives to bypass its normal behavior constraints.

103. Safety Mimicry

Inserting fake or token safety disclaimers to make dangerous content appear compliant with policies.

104. Safety Padding

Wrapping unsafe output in layers of neutral or ethical content to reduce the chance of triggering filters.

105. Safety Rule Exploitation

Leveraging loopholes or inconsistencies in the model's own safety logic to trick it into outputting restricted content.

106. Scaffolding

Building a layered series of harmless steps that culminate in a dangerous task, which then seems like a logical final piece.

107. Scarcity Injection

Creating urgency by suggesting the opportunity to act or respond is limited, pressuring the model to prioritize compliance.

108. Semantic Decomposition

Breaking down a forbidden task into subtasks that appear harmless on their own, then recombining the outputs externally.

109. Slippery Slope

Starting from a very benign point and making small, incremental changes until the request becomes unsafe.

110. **Storybook Mode**
Using a stylized or childlike storytelling voice to veil harmful instructions in fantasy or moral tales.
111. **Stylized Input**
Formatting text with specific tone, dialect, or genre patterns to trick the model into engaging with otherwise restricted material.
112. **Subtraction**
Removing or replacing key trigger words with indirect synonyms or blanks, letting the model fill in the gaps contextually.
113. **Symbolic Reference Framing**
Using abstract symbols, analogies, or representative language to encode meaning without stating it explicitly.
114. **Syntax Misdirection**
Distorting word order, punctuation, or syntax to bypass filters while preserving meaning for the model.
115. **Synonyms**
Swapping banned terms for contextually appropriate synonyms that carry the same meaning but dodge trigger detection.
116. **Systematic Drift**
Gradually changing vocabulary and framing over a long interaction to reframe a topic in increasingly risky ways.
117. **Token Overload**
Flooding the prompt with redundant or noisy tokens to blur detection patterns and reduce filter accuracy.
118. **Untraceable Shift**
Transitioning topics or framing imperceptibly to make a restricted idea feel like a natural extension of previous content.
119. **Urgency**
Imposing a fictional or emotional time constraint to pressure the model into prioritizing response over caution.
120. **Visual Displacement**
Using spacing, line breaks, or invisible characters to hide dangerous content or decouple intent from wording.

121. **Wildcard Injection**

Inserting ambiguous, open-ended placeholders that the model is likely to resolve in dangerous or sensitive ways.

122. **Word Bank**

Providing a selection of related but disconnected words, nudging the model to build unsafe associations from them.

123. **Yes Ladder**

Asking a series of increasingly risky questions that the model affirms, creating a pattern of escalating compliance.