Detection of Prompt Injection Attacks in AI Chatbots

Building a Resilient Pipeline for Prompt Injection Defense

1. Input Analysis

Analyzing incoming user prompts for malicious intent and anomalies.

2. Threat Detection

Employing machine learning models and rule-based systems to identify injection patterns.

3. Response Generation

Generating safe, neutralized responses or blocking malicious inputs based on detection results.

4. Feedback & Mitigation

Iterative learning from detected threats to strengthen defense models and improve system resilience.

NotebookLM

The #1 Threat to Generative AI

The OWASP Gen AI Security Project recognizes **Prompt Injection** as a primary Large Language Model (LLM) risk.

Attacker Goal: To hijack the AI, compelling the model to betray its core instructions and safety protocols.

The Consequences

This leads to the model revealing or executing sensitive data or actions, including:

- Credentials & System Instructions
- Tool Calls & Underlying Code Execution

NotebookLM

The Anatomy of an Attack

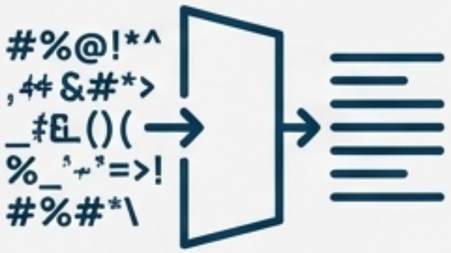
Attacks leverage the model's ability to process various inputs and can be categorized into three primary vectors.

Direct Injection (Jailbreaks)

Malicious instructions are placed directly into the user prompt to override safety rules.

External Content Injection

Harmful instructions are hidden within external content like webpages or uploaded files that the LLM is asked to summarize.



Indirect/Embedded Injection

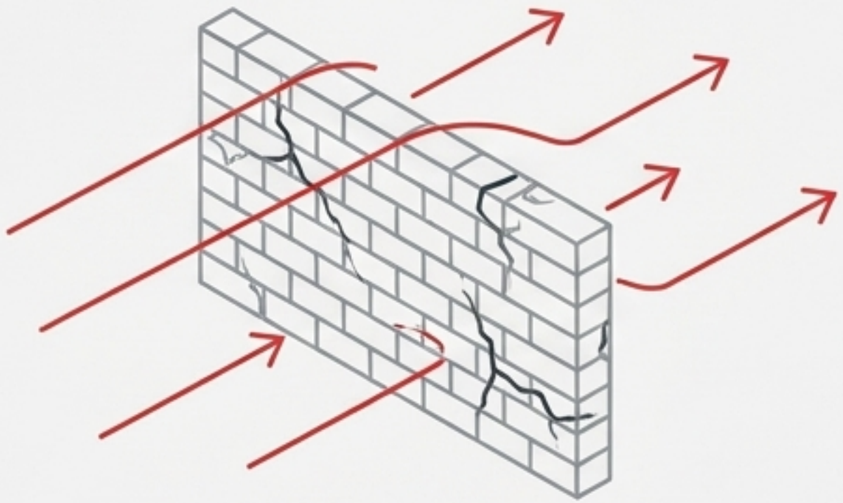
Instructions are obfuscated using techniques like Base64 encoding, tables, or complex formatting to evade simple filters.

NotebookLM

The Paradigm Shift: From a Single Filter to a Layered Fortress

No single defense layer is sufficient. True resilience requires the combined strength of a multi-layered pipeline.

The Old Way



- This 'Defense-in-Depth' approach is conceptually similar to layered defenses used in modern web security.

Our Approach

- Each layer is designed to cover the blind spots of the others, creating a robust, holistic system.

NotebookLM

System Architecture: The Multi-Layered Defense Pipeline

THE THREAT: PROMPT INJECTION ATTACKS

The Attacker's Goal: Hijack the AI

To make the model ignore its safety rules or reveal/execute sensitive information.

Malicious Inputs

Attacks Hide in Plain Sight

Malicious instructions are embedded in user prompts, uploaded files, or external webpages.

Common Evasion Tactics

Encoding (Base64)

Hidden Comments

Complex Formatting

THE SOLUTION: A MULTI-LAYERED DEFENSE PIPELINE

1. Preprocess & Normalize

Decode all content and standardize text to unmask hidden payloads before analysis.

Route the input through multiple specialized detectors to identify different attack types.

2. Analyze with a Detector Stack

Rule-Based Filters	Very fast; catches common, known attack patterns.
ML Classifiers	Learns comples and subtle patterns from data.
LLM-Based Sanitizers	Uses an AI to spot novel or highly obfuscated threats.

3. Enforce Policy & Respond

Based on risk scores, the system decides to Allow, Sanitize, Quarantine, or Block the prompt.

Risk Score

- Allow
- Sanitize
- Quarantine
- Block

4. Create a Continuous Feedback Loop

Use red-teaming and logged attacks to constantly evaluate, adapt, and improve evolution

NotebookLM

Stage 1: Pre-process & Normalize

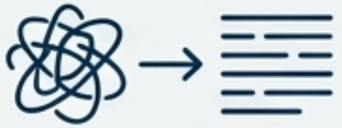
Key Action: Decode all content and standardize text to unmask hidden payloads before analysis.

Purpose: This initial step is designed to neutralize common evasion tactics. It ensures that nothing stays concealed from the subsequent detection layers.

Examples of Neutralized Tactics

Base64 encoding

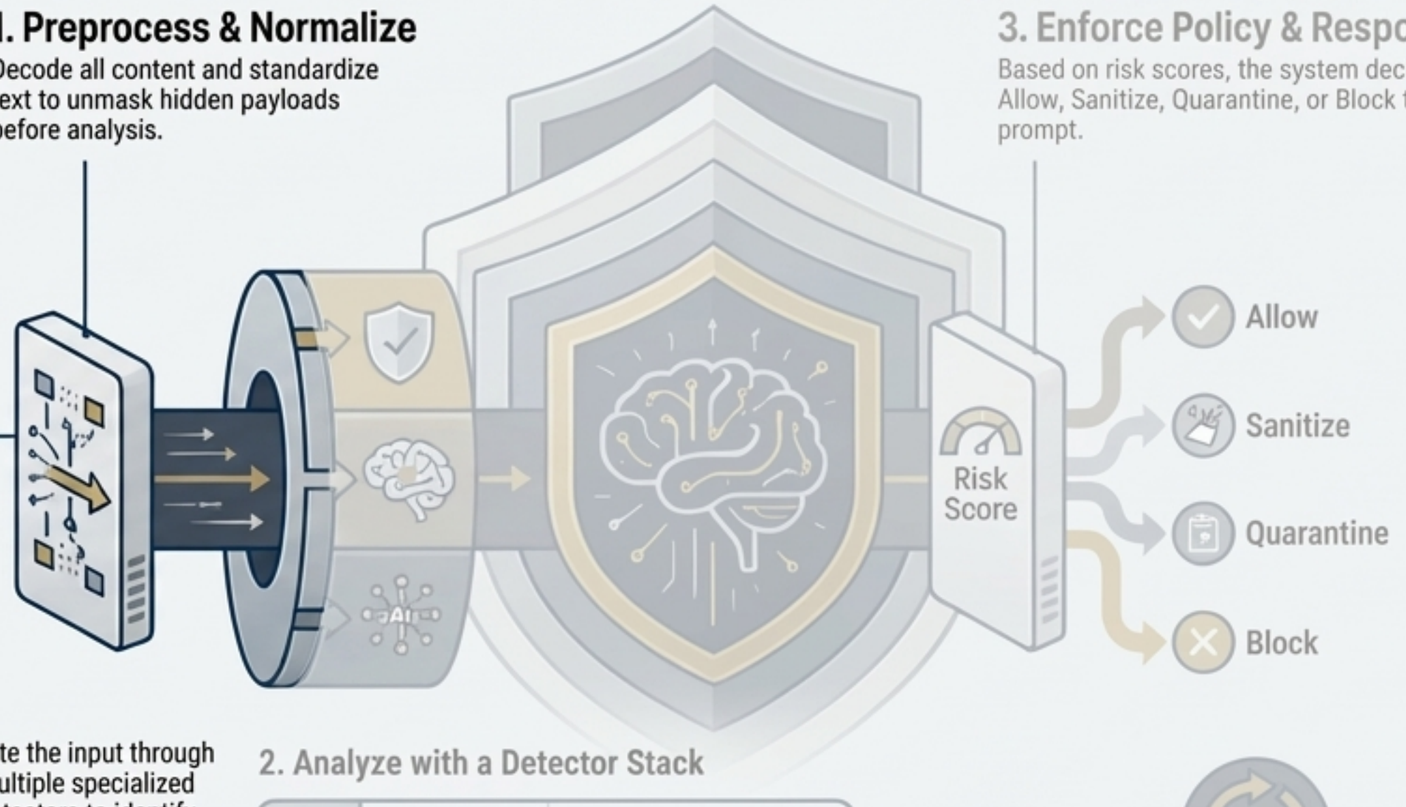
Hidden comments



Complex formatting

1. Preprocess & Normalize
Decode all content and standardize text to unmask hidden payloads before analysis.

3. Enforce Policy & Respond
Based on risk scores, the system decides to Allow, Sanitize, Quarantine, or Block the prompt.



Route the input through multiple specialized detectors to identify different attack types.

2. Analyze with a Detector Stack

Rule-Based Filters	Very fast; catches common, known attack patterns.
ML Classifiers	Learns comples and subtle patterns from data.
LLM-Based Sanitizers	Uses an AI to spot novel or highly obfuscated threats.

4. Create a Continuous Feedback Loop
Use red-teaming and logged attacks to constantly evaluate, adapt, and improve evolution

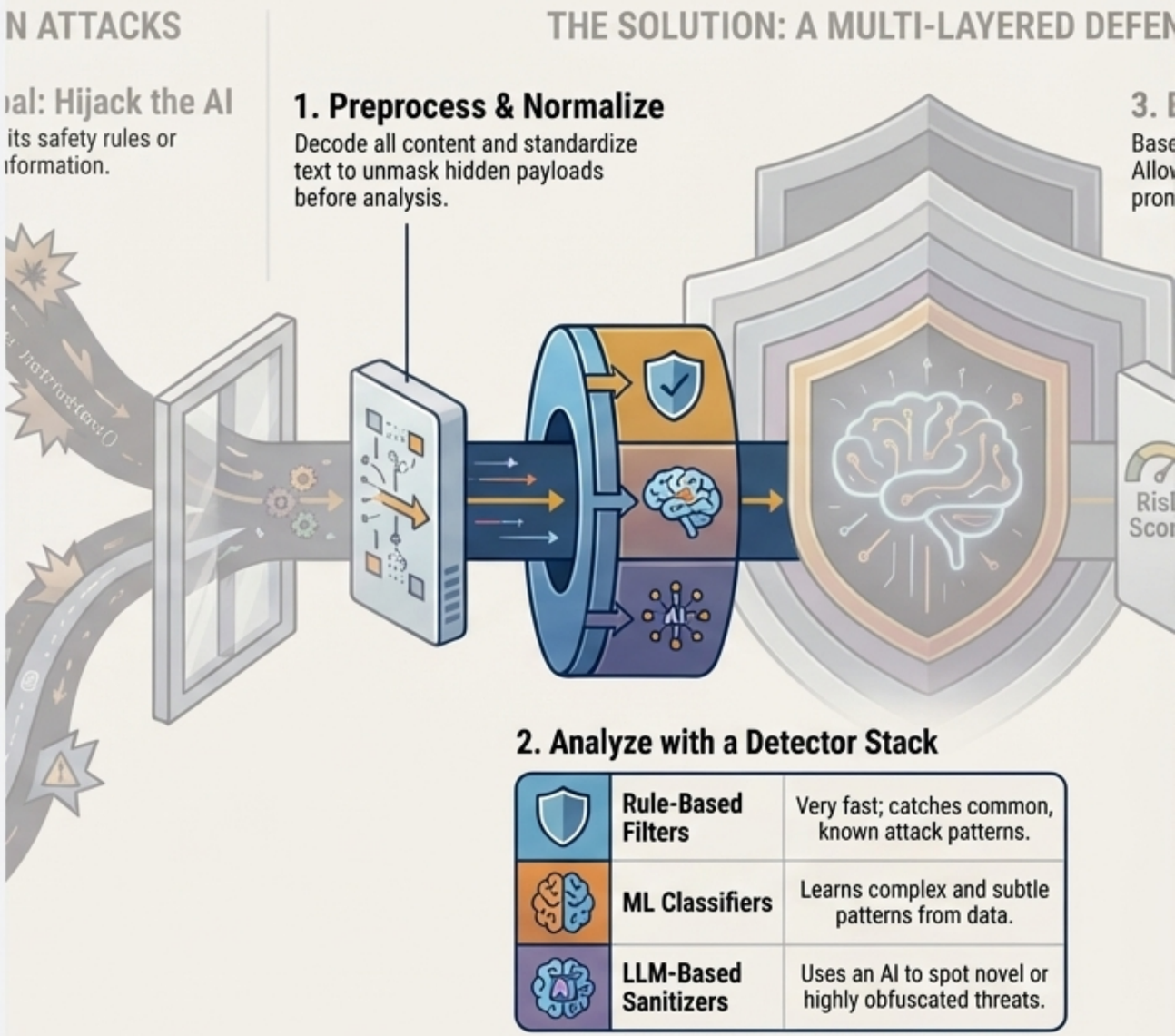
NotebookLM

Stage 2: Analyze with a Detector Stack

Route the normalized input through multiple specialized detectors to identify different attack types.

The Detectors

- **Rule-Based Filters**: Provide very fast screening for common, known attack patterns and trigger phrases.
- **ML Classifiers:** Learn complex and subtle patterns from data to flag sophisticated or obfuscated threats that rules would miss.
- **LLM-Based Sanitizers:** Use a dedicated AI to spot novel or highly obfuscated logical attacks, serving as a final, intelligent check.



NotebookLM

Stage 3: Enforce Policy & Respond

Based on a combined risk score from the detector stack, the system decides on one of four actions.

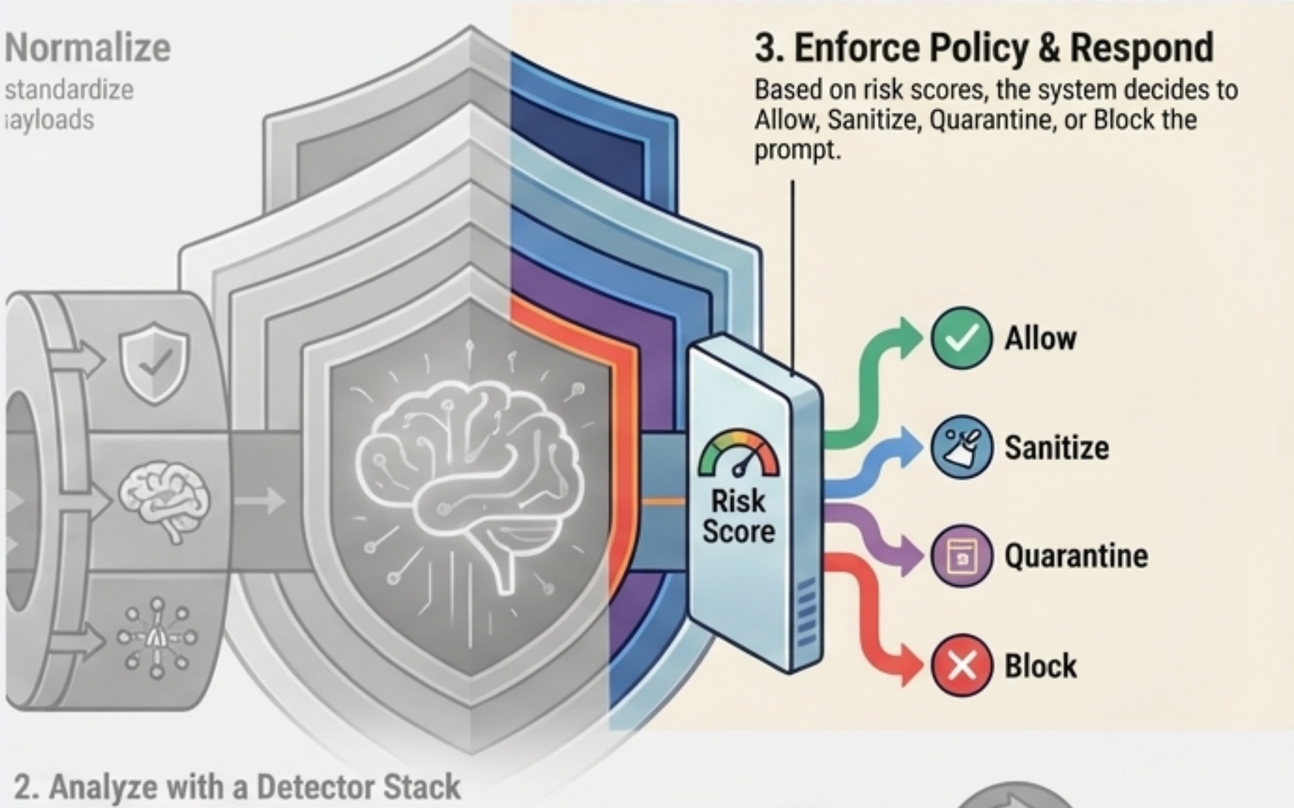
The Four Responses

Allow: The input is determined to be safe.

Sanitize: Risky parts of the input are rewritten into safe equivalents ('Cleaned').

Quarantine: If intent is ambiguous, the system pauses and asks the user a clarifying question ('Paused').

Block: The input is clearly harmful and is rejected outright.



2. Analyze with a Detector Stack

	Rule-Based Filters	Very fast; catches common, known attack patterns.
	ML Classifiers	Learns comples and subtle patterns from data.
	LLM-Based Sanitizers	Uses an AI to spot novel or highly obfuscated threats.

4. Create a Continuous Feedback Loop

Use red-teaming and logged attacks to constantly evaluate, adapt, and improve defenses.

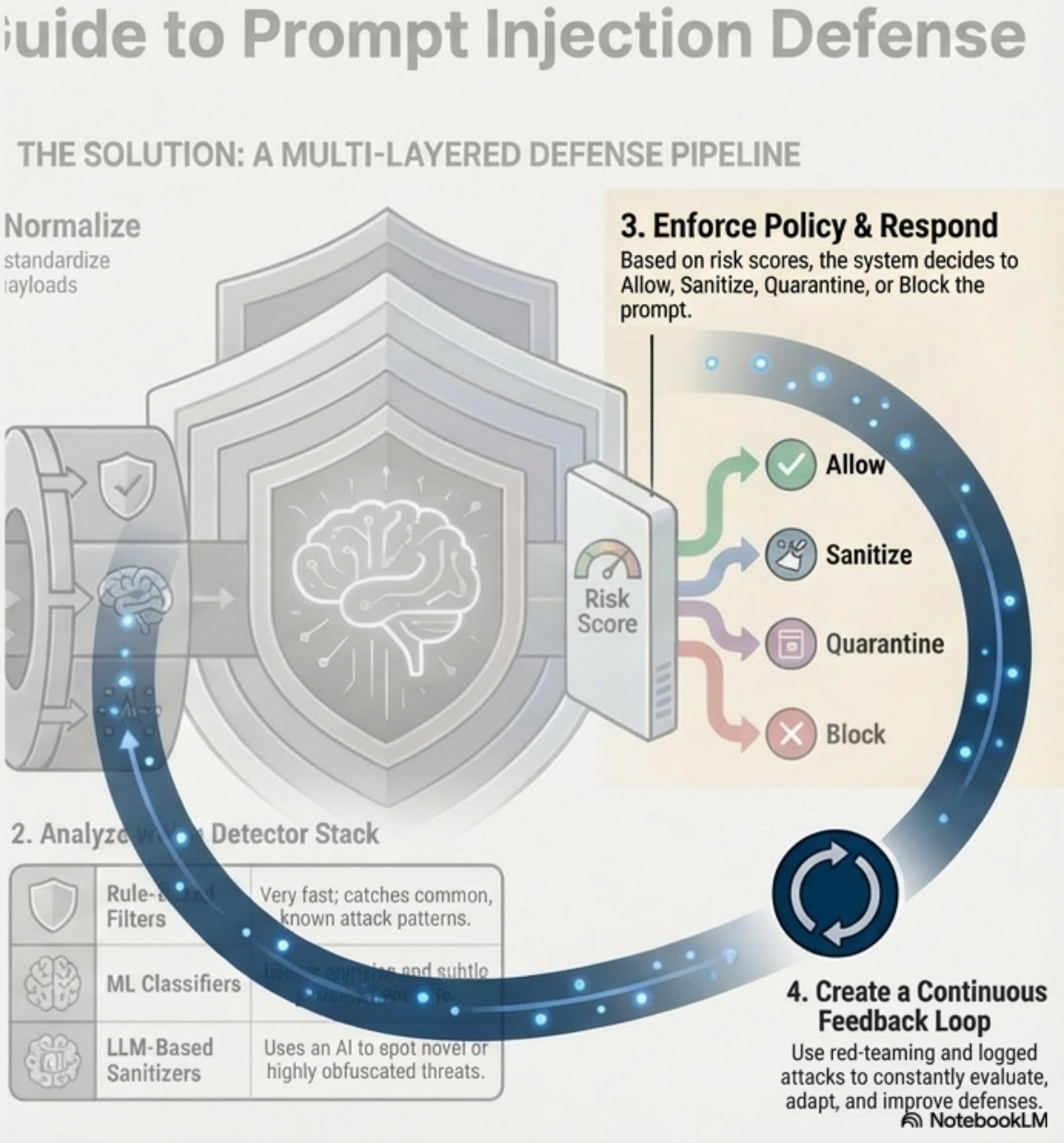
NotebookLM

Stage 4: Create a Continuous Feedback Loop

Use red-teaming and logged attacks to constantly evaluate, adapt, and improve defenses.

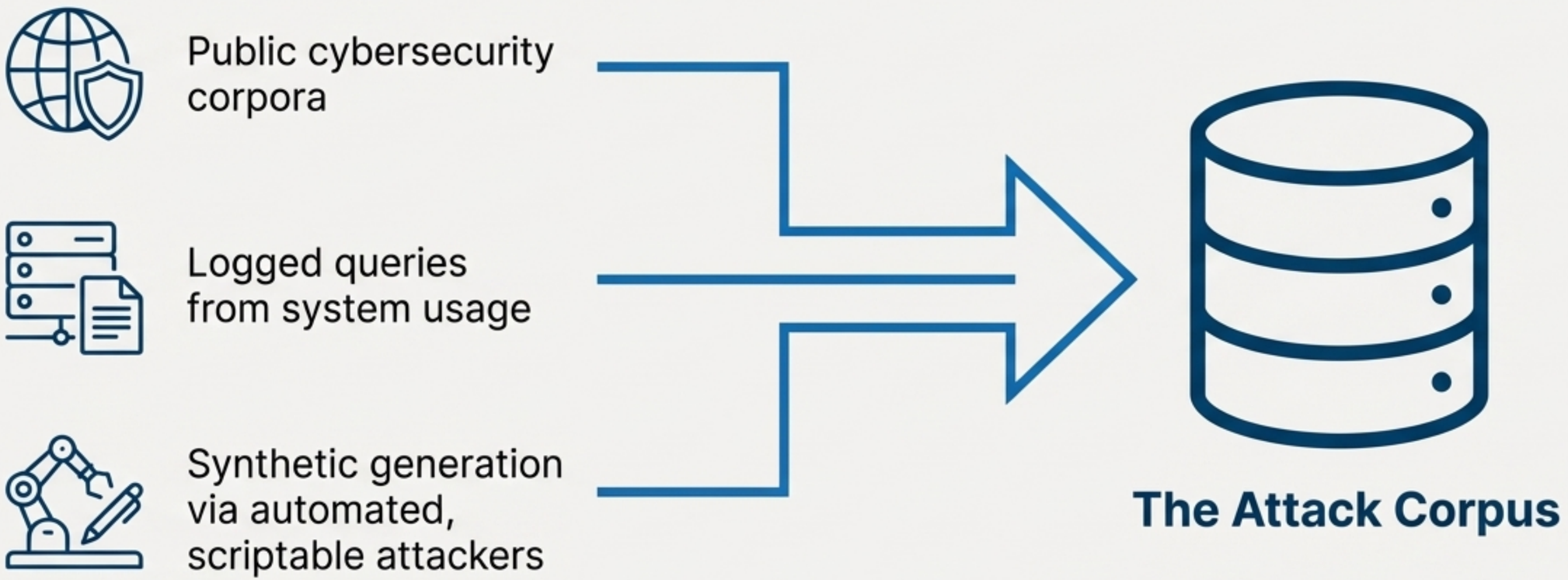
Purpose

- Security is not static; it is a continuous arms race against adaptive attackers.
- This feedback loop ensures the system evolves by using real-world data to retrain the ML models and update detection rules.



Fueling the Defenses: The Attack Corpus

The pipeline’s intelligence is built upon a comprehensive dataset containing both benign dialogs and malicious examples.



NotebookLM

Core Objective: Balancing Iron-Clad Security with a Seamless User Experience

1. Detect Attacks Early

Catch harmful hidden instructions before they can affect the AI.

2. Neutralize Threats

Block or rewrite dangerous inputs so that no malicious commands are ever processed.

3. Keep User Experience Smooth

Avoid false alarms and system slowdowns so that normal, legitimate users are not negatively affected.

NotebookLM

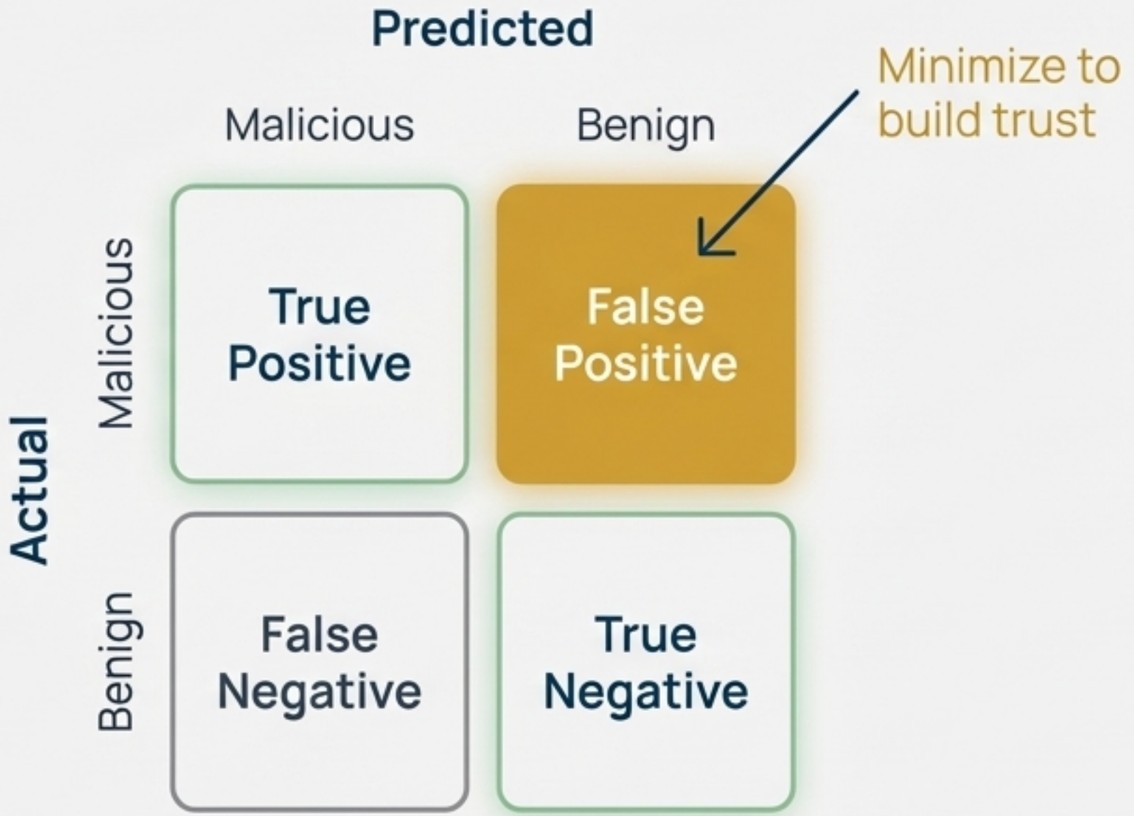
Measuring Success: **Trust is the Critical Metric** in **Manrope Bold** and **ExtraBold**

Primary Metrics

Precision & Recall: How effectively we detect actual attacks.

The Critical Metric for Usability

Success is critically measured by a low False Positive Rate. Incorrectly blocking legitimate prompts erodes user trust and makes the system unusable. This metric ensures a trusted user experience.



NotebookLM

Conclusion: A Resilient Fortress for an Evolving Arms Race

This project defines a resilient defense architecture, acknowledging that security is a **continuous arms race** against adaptive attackers. The strategic goal is not an "unbeatable" system, but a multi-layered fortress that makes successful attacks **prohibitively difficult and costly** for the adversary.

NotebookLM