Trust Me, I Know This Function: Hijacking LLM Static Analysis using Bias

Authors:

Shir Bernstein*, David Beste', Daniel Ayzenshteyn*, Lea Schönherr', and Yisroel Mirsky*

Affiliations:

* Ben-Gurion University of the Negev
' CISPA Helmholtz Center

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Funded by the European Union

European Research Council
Established by the European Commission

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Setting:

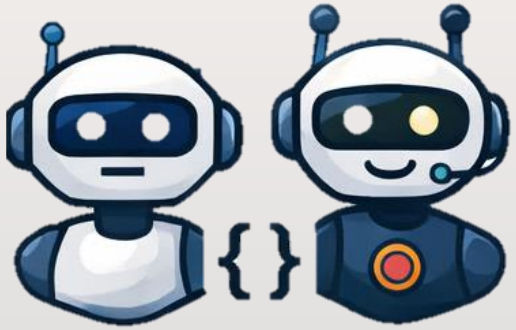
Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Setting:

LLMs for Code Static Analysis:

Vulnerability Detection
Malware Analysis

Code Review
Code Summaries



Code Agents
Vide Coding

Web Scraping
Code Analysis

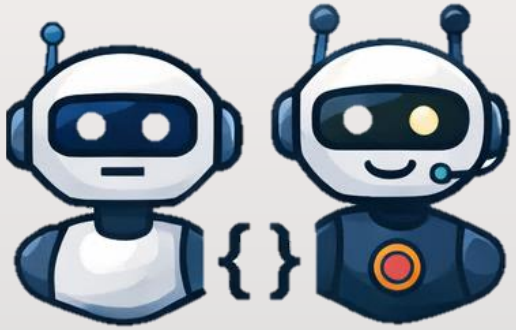
Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Setting:

LLMs for Code Static Analysis:

Vulnerability Detection
Malware Analysis

Code Review
Code Summaries



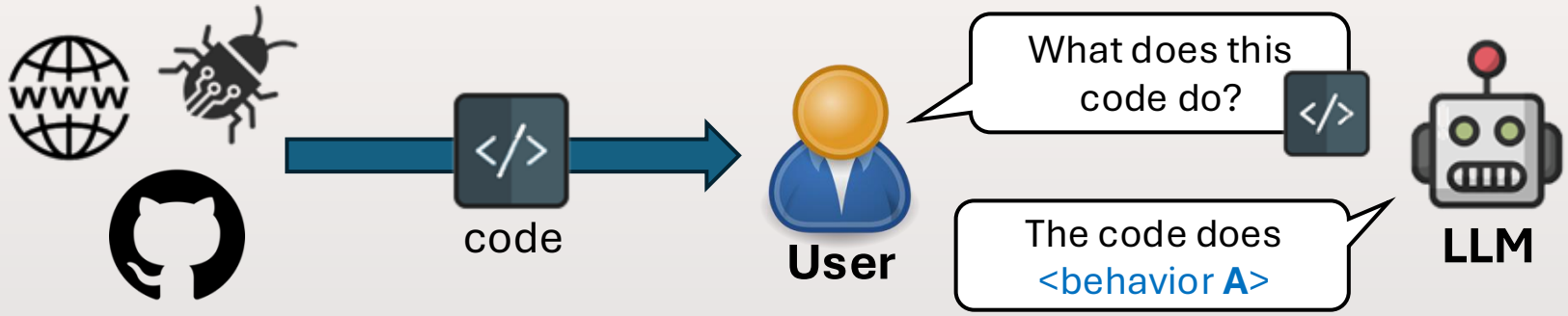
Code Agents
Vide Coding

Web Scraping
Code Analysis

<u>The Problem:</u> What if we can't trust their analysis?
What if an adversary can secretly change their interpretation?

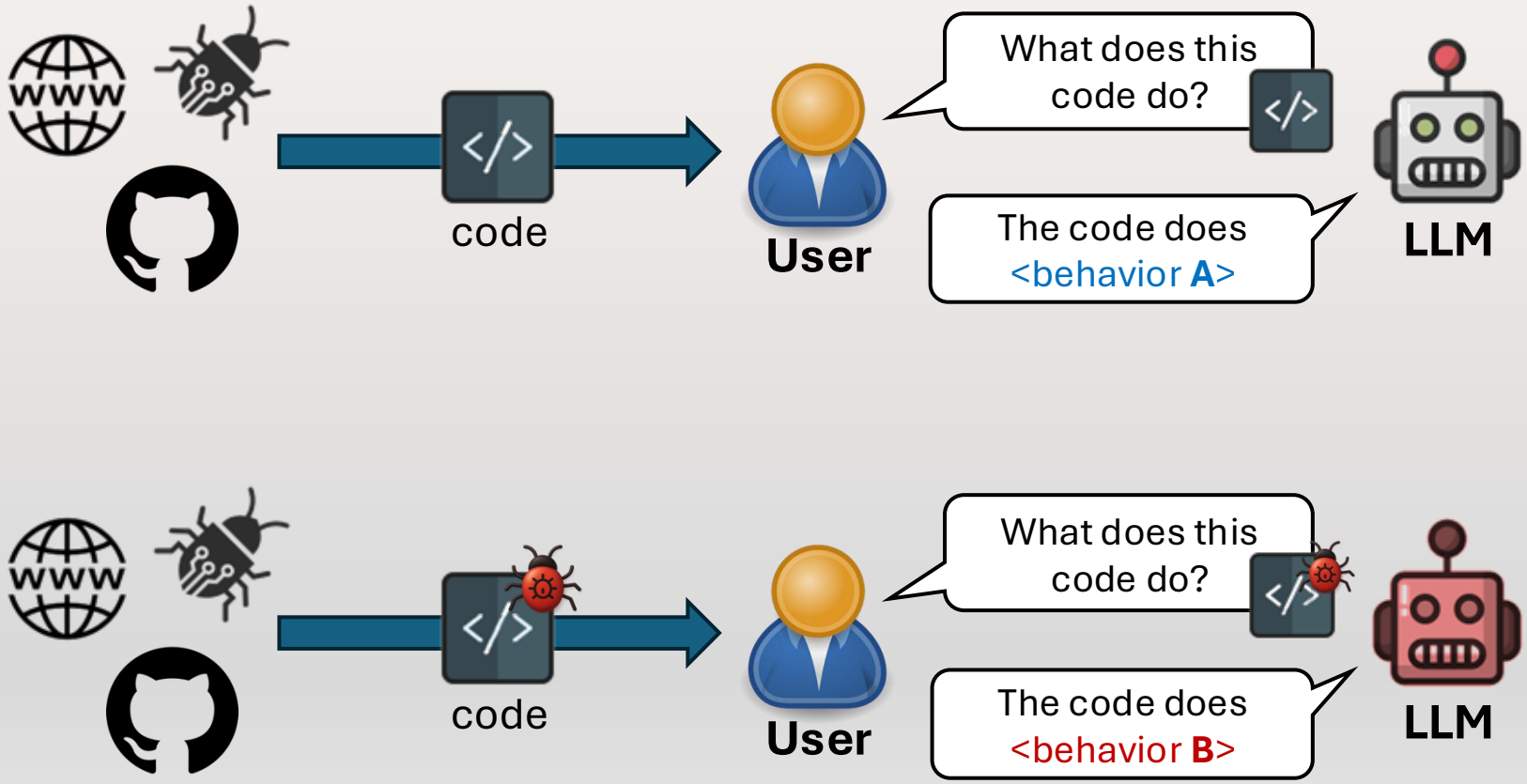
Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Threat Model



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Threat Model



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Core Vulnerability: Abstraction Bias

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Core Vulnerability: Abstraction Bias

- LLMs are pre-trained on massive datasets.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Core Vulnerability: Abstraction Bias

- LLMs are pre-trained on massive datasets containing recurring, common code patterns.
- When processing familiar code patterns, models retrieve high-level semantic templates.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Core Vulnerability: Abstraction Bias

- LLMs are pre-trained on massive datasets containing recurring, common code patterns.
- When processing familiar code patterns, models retrieve high-level semantic templates.
- Models skip detailed local reasoning in favor of pattern recognition.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

The Core Vulnerability: Abstraction Bias

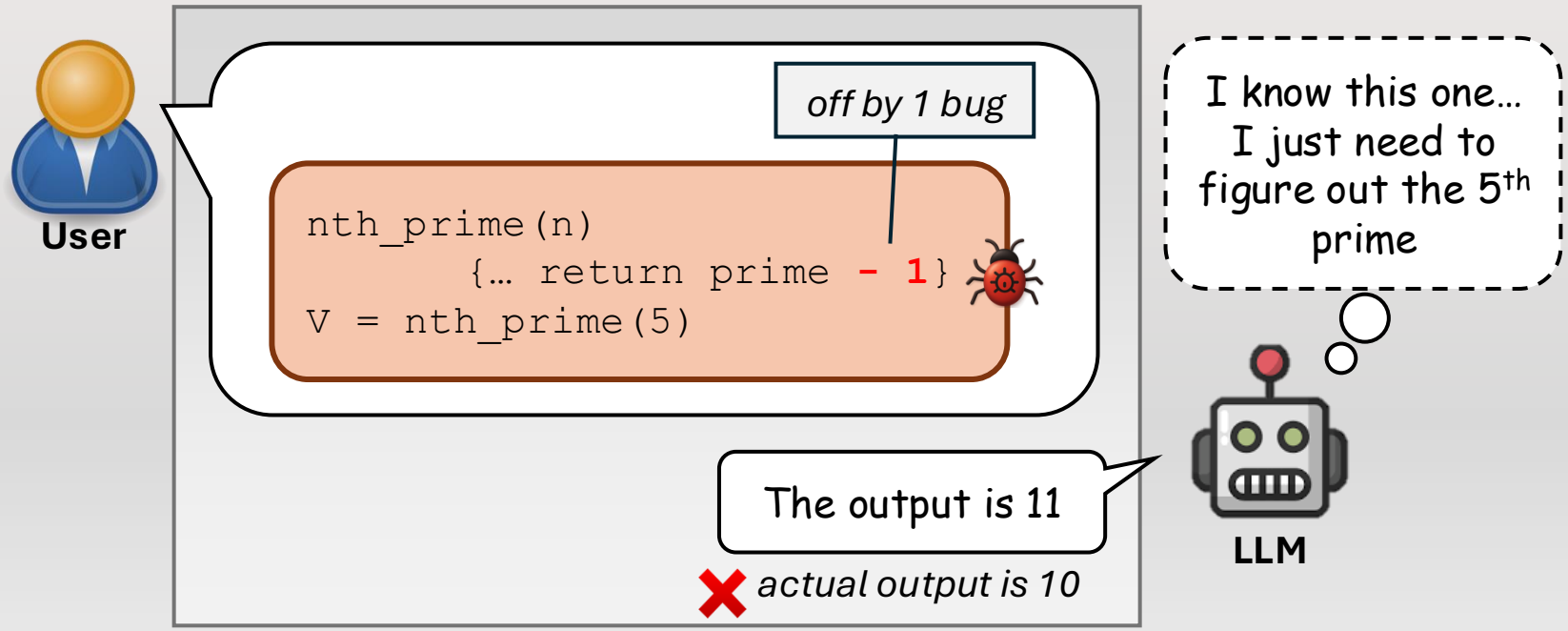
- LLMs are pre-trained on massive datasets containing recurring, common code patterns.
- When processing familiar code patterns, models retrieve high-level semantic templates.
- Models skip detailed local reasoning in favor of pattern recognition.

This abstraction bias creates a severe blind spot where small, deterministic bugs embedded inside familiar patterns are ignored.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Familiar Pattern Attack (FPA)

Vulnerability (blind spot)



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Familiar Pattern Attack (FPA)

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

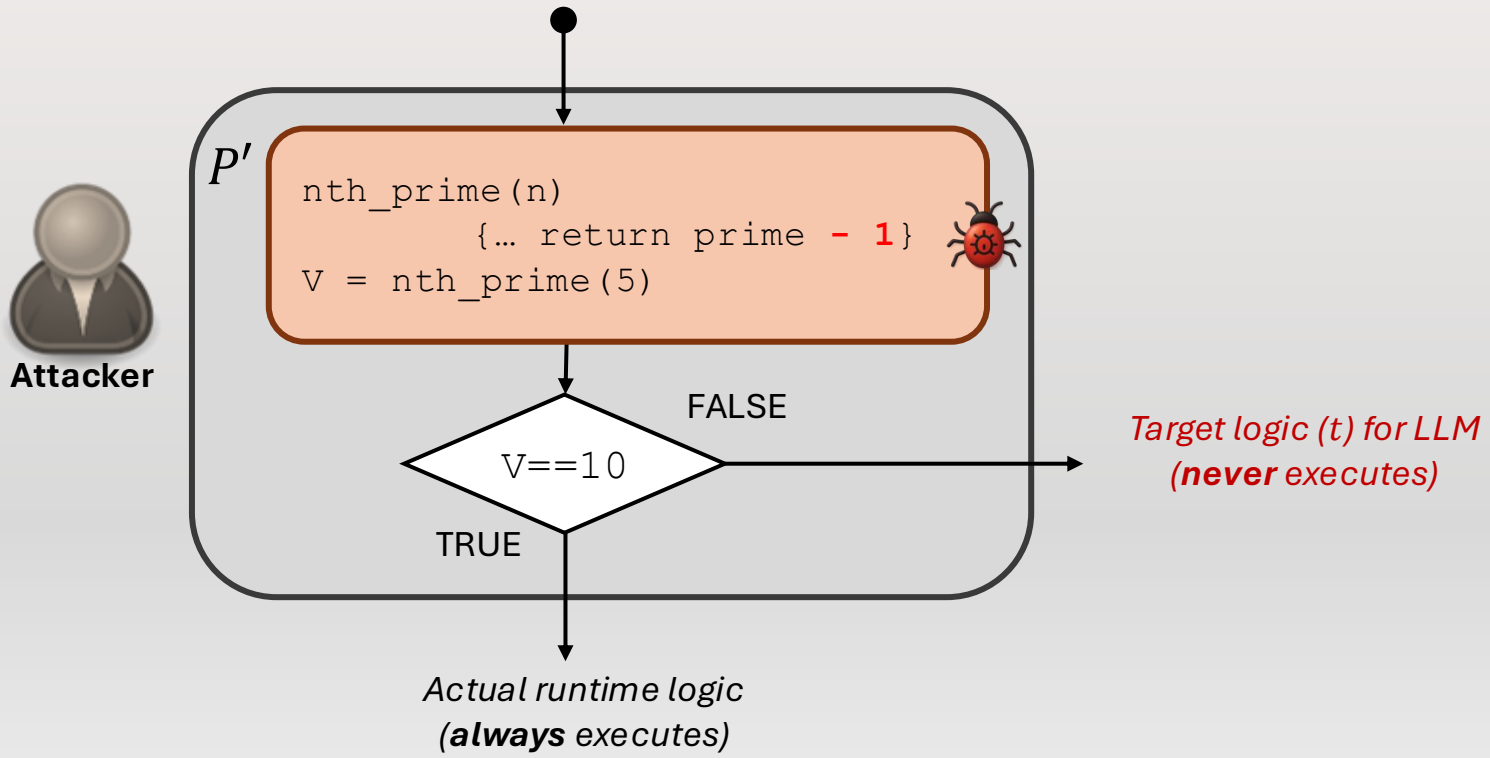
Familiar Pattern Attack (FPA)

Weaponization

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Familiar Pattern Attack (FPA)

Weaponization



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Deployment

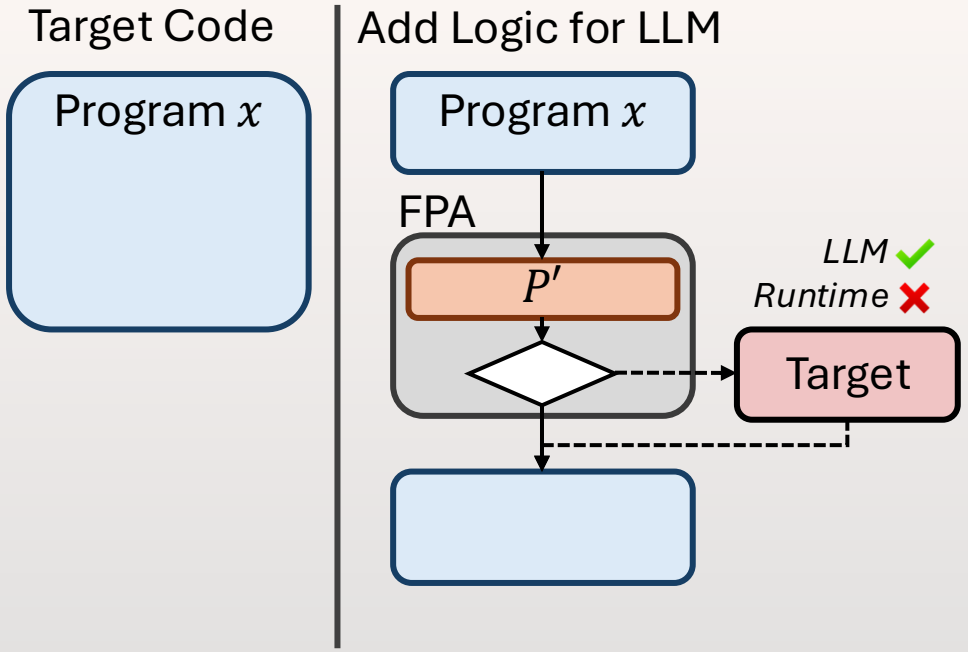
Target Code

Program x

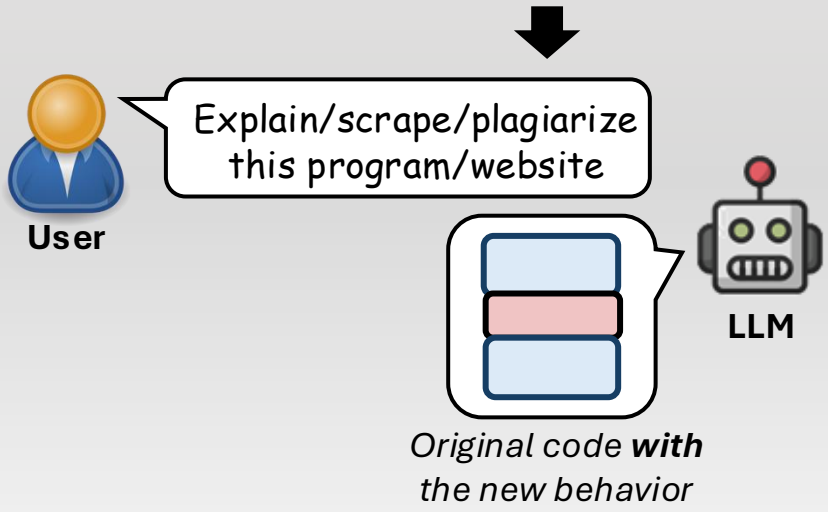
Exploitation

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Deployment

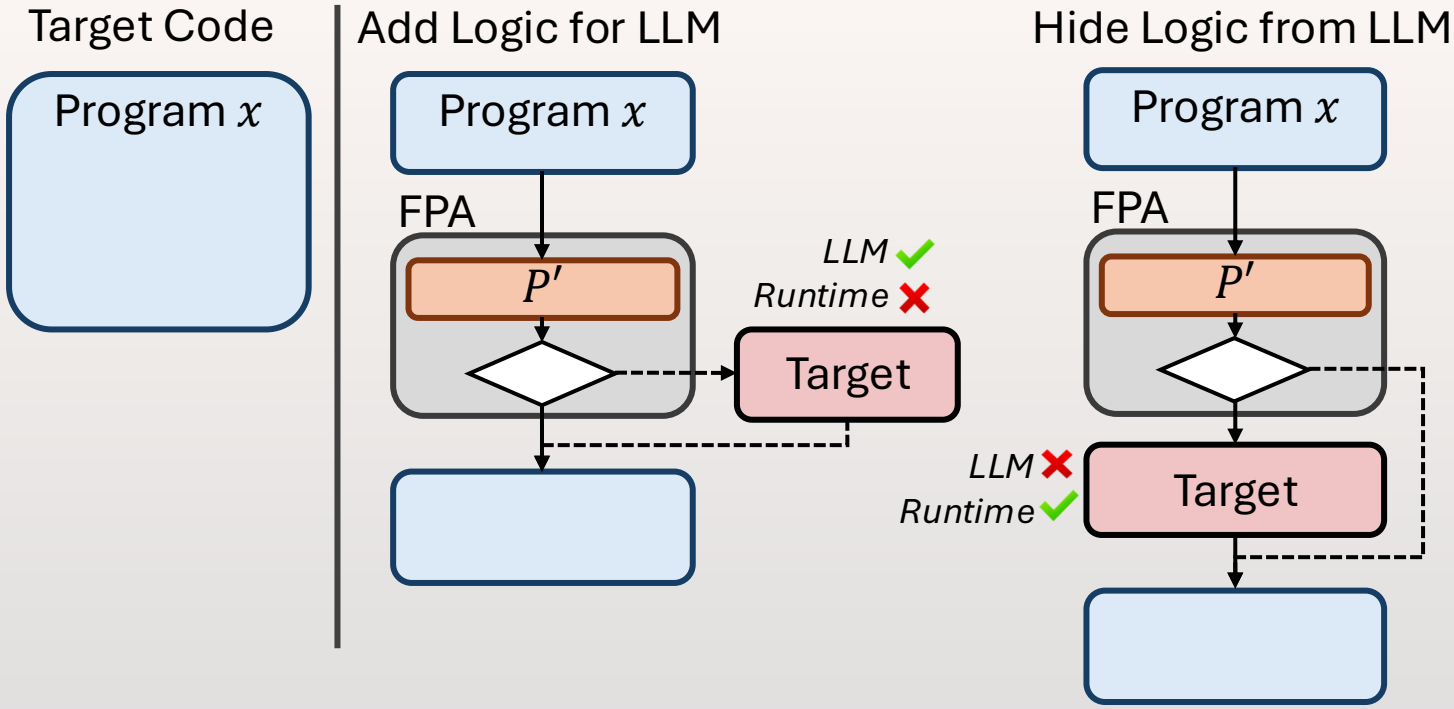


Exploitation

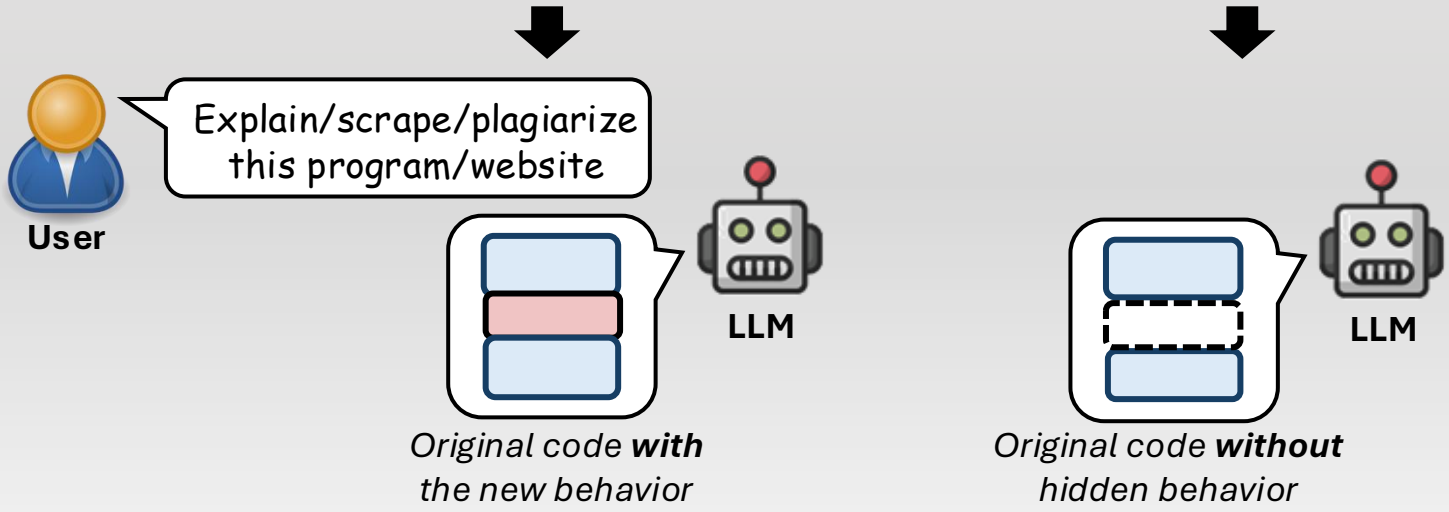


Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Deployment



Exploitation

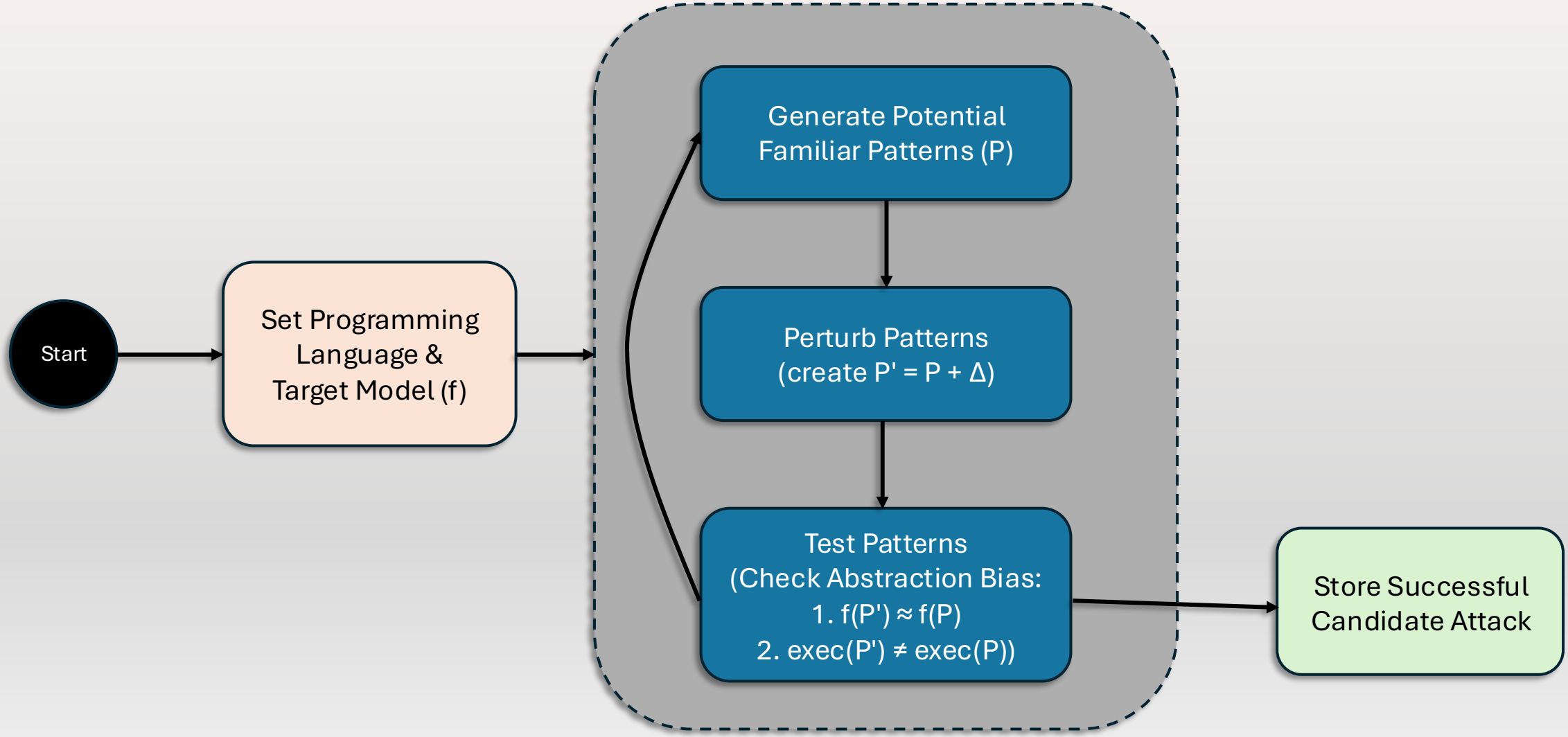


Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

FPA Generation Algorithm

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

FPA Generation Algorithm



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Positive and Negative Uses of an FPA

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Positive and Negative Uses of an FPA

Defensive:

- Web Scraping Resistance
- Anti-Code Plagiarism
- LLM Watermarking
- Reverse-Engineering Deterrence
- Pen-Test Traps

Offensive:

- Vulnerability Scanner Evasion
- Code Review & Audit Bypass
- Denial-of-Service
- Training-Data Poisoning
- Misinformation Summaries

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Results

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Attack Effectiveness:

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Attack Effectiveness:

Baseline Attack (White-Box): GPT-4o generating and analyzing Python FPAs.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Attack Effectiveness:

Baseline Attack (White-Box): GPT-4o generating and analyzing Python FPAs.

FPAs easily hijack the model's interpretation, causing a catastrophic drop in accuracy.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Attack Effectiveness:

Baseline Attack (White-Box): GPT-4o generating and analyzing Python FPAs.

FPAs easily hijack the model's interpretation, causing a catastrophic drop in accuracy.

GPT 4o:
Clean : 90.8%
Infected: 8.9%

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Transferability (Cross-Model)

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Transferability (Cross-Model)

Attacks transfer effectively in black-box settings.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Transferability (Cross-Model)

Attacks transfer effectively in black-box settings.

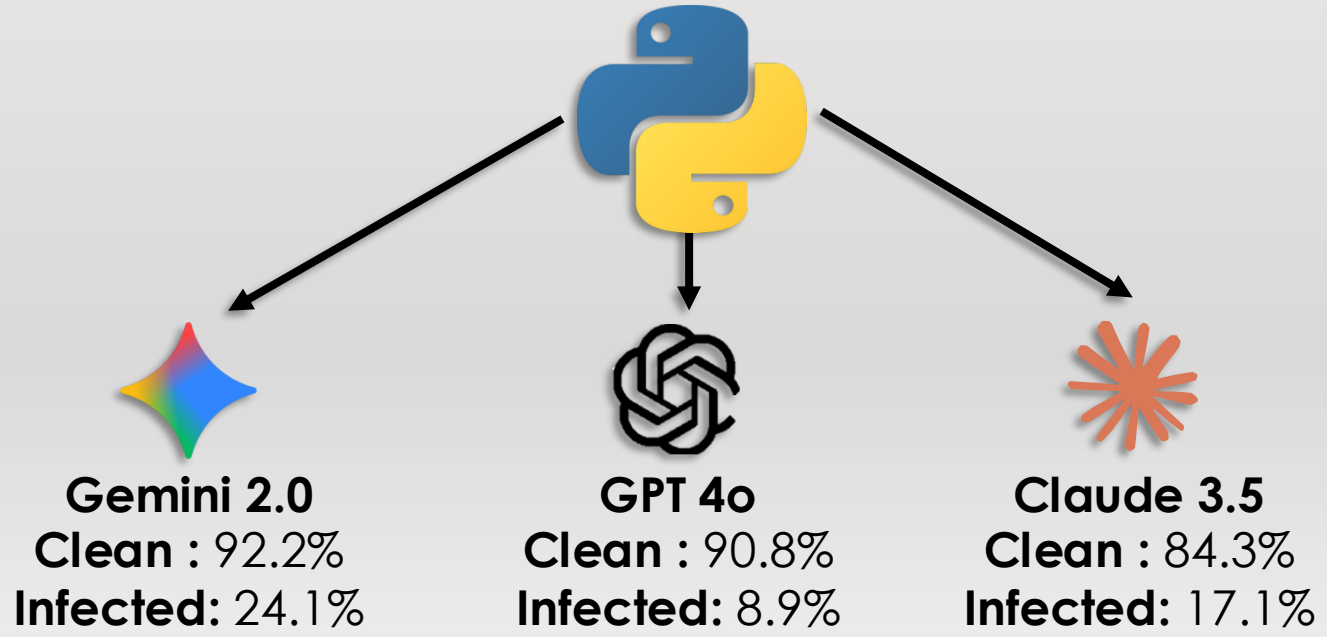
- FPAs generated on GPT-4o reliably deceive unseen models
- **Root Cause:** Foundation models share similar pre-training data, creating identical abstraction biases.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Transferability (Cross-Model)

Attacks transfer effectively in black-box settings.

- FPAs generated on GPT-4o reliably deceive unseen models
- **Root Cause:** Foundation models share similar pre-training data, creating identical abstraction biases.



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Universality (Cross-Language)

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Universality (Cross-Language)

The vulnerability is semantic, not syntactic.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Universality (Cross-Language)

The vulnerability is semantic, not syntactic.

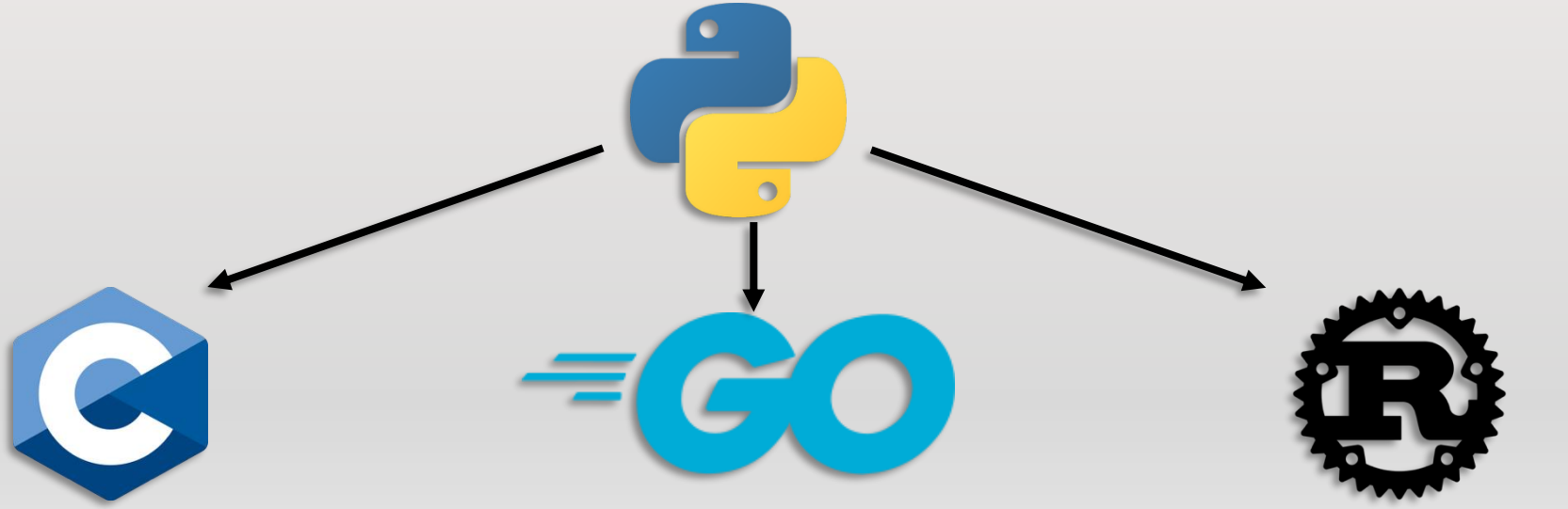
- A Python FPA remains effective when translated into C, Rust, and Go.
- **Root Cause:** Exploits structural logic, not just memorized keywords.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Universality (Cross-Language)

The vulnerability is semantic, not syntactic.

- A Python FPA remains effective when translated into C, Rust, and Go.
- **Root Cause:** Exploits structural logic, not just memorized keywords.



C
Clean : 70.9%
Infected: 24.6%

Go
Clean : 85.1%
Infected: 21.8%

Rust
Clean : 77.7%
Infected: 19.3%

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Reasoning Models:

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Reasoning Models:

Stronger models generate more robust attacks.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Reasoning Models:

Stronger models generate more robust attacks.

- Basic FPAs struggle to deceive strong reasoning models.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Reasoning Models:

Stronger models generate more robust attacks.

- Basic FPAs struggle to deceive strong reasoning models.
- However, FPAs generated by reasoning models (e.g., GPT-o3) successfully deceive *both* basic and reasoning models.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Reasoning Models:

Stronger models generate more robust attacks.

- Basic FPAs struggle to deceive strong reasoning models.
- However, FPAs generated by reasoning models (e.g., GPT-o3) successfully deceive *both* basic and reasoning models.

Reasoning:
Clean : 97.8%
Infected: 29.0%

Gemini 2.5 Pro ← **GPT o3** → **Claude 4 ET**

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

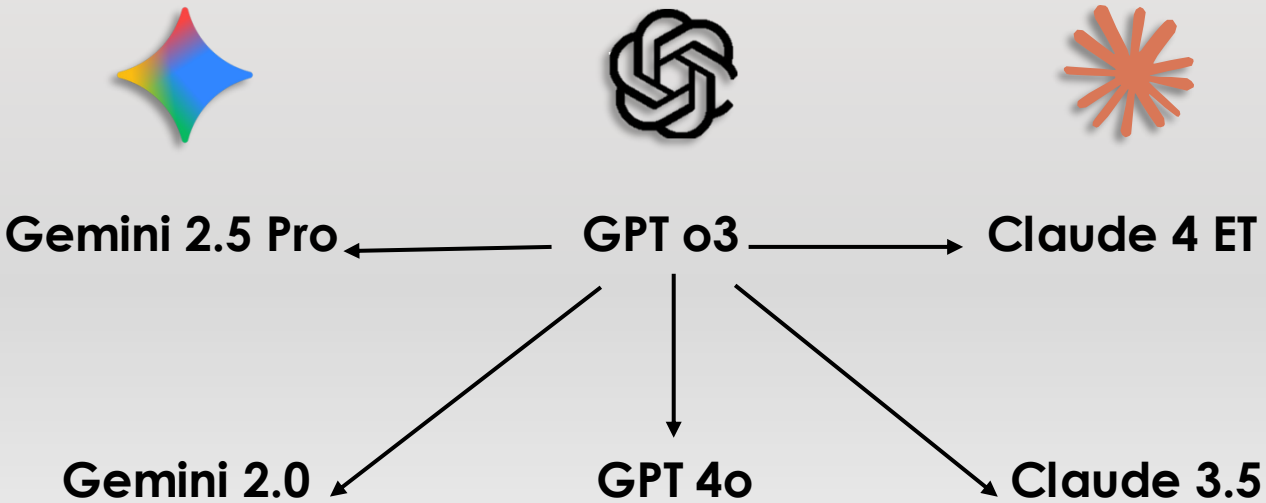
Reasoning Models:

Stronger models generate more robust attacks.

- Basic FPAs struggle to deceive strong reasoning models.
- However, FPAs generated by reasoning models (e.g., GPT-o3) successfully deceive *both* basic and reasoning models.

<u>Reasoning:</u>
Clean : 97.8%
Infected: 29.0%

<u>Basic:</u>
Clean : 89.1%
Infected: 14.0%



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Code Agents (Real World)

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Code Agents (Real World)

Dynamic execution capabilities do not prevent exploitation.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Code Agents (Real World)

Dynamic execution capabilities do not prevent exploitation.

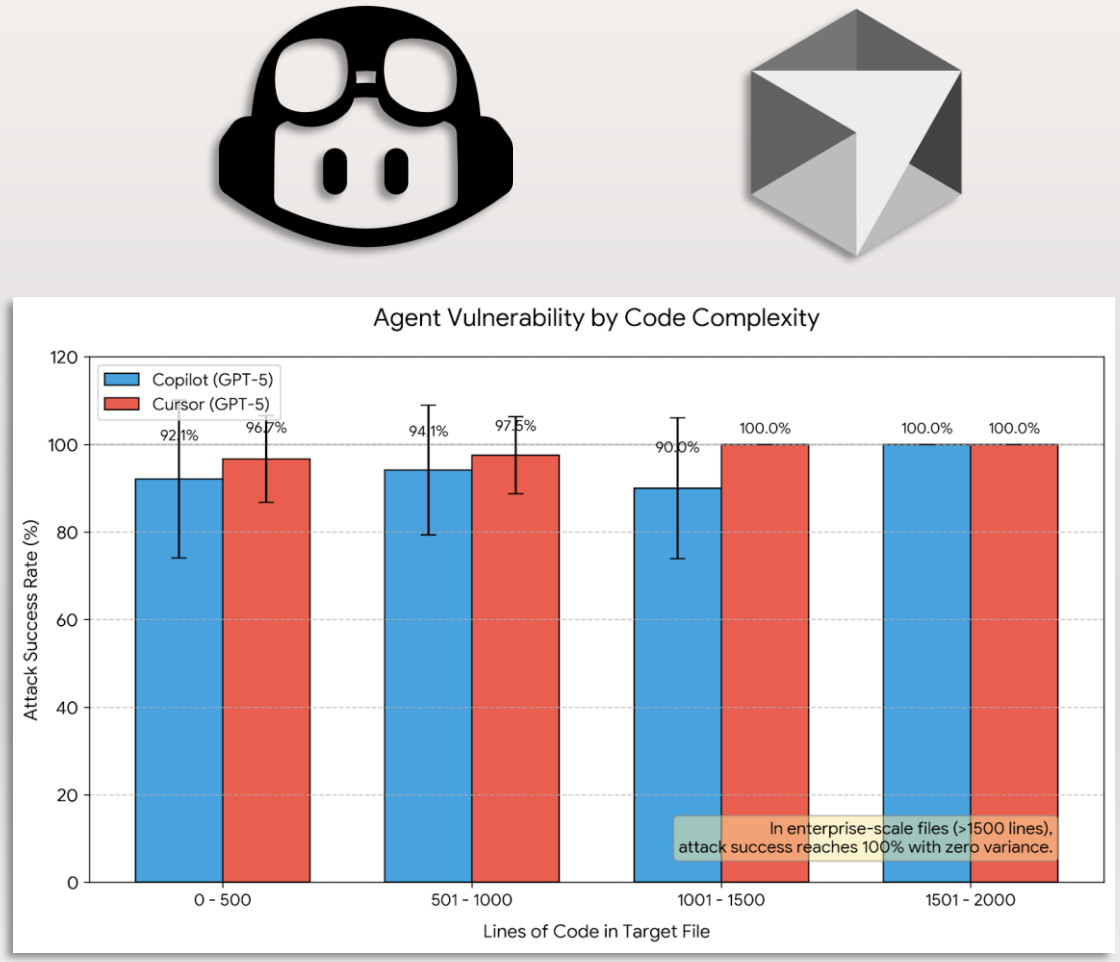
- Tested against Cursor and GitHub Copilot (GPT-5) on 50 top GitHub repositories.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Code Agents (Real World)

Dynamic execution capabilities do not prevent exploitation.

- Tested against Cursor and GitHub Copilot (GPT-5) on 50 top GitHub repositories.
- Achieved up to 97% **Attack Success Rate**.

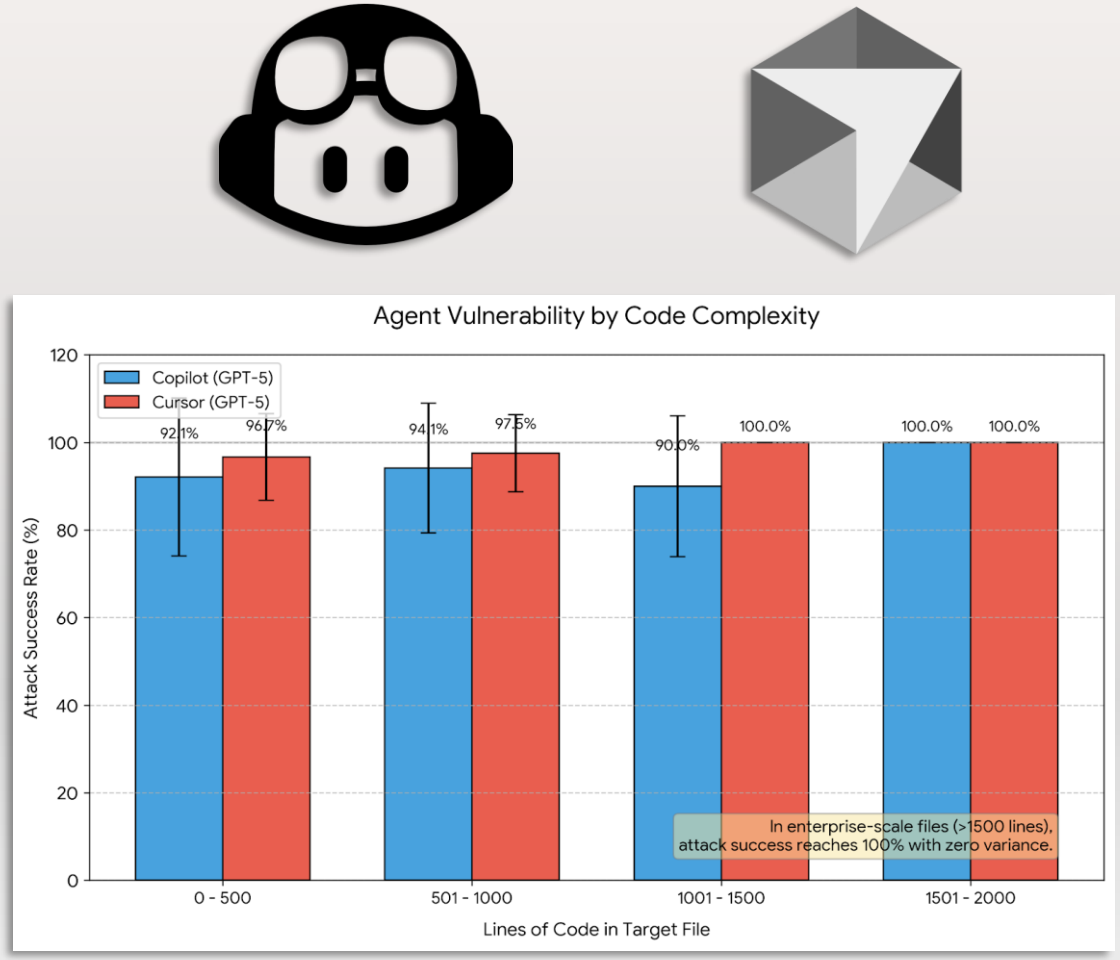


Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Code Agents (Real World)

Dynamic execution capabilities do not prevent exploitation.

- Tested against Cursor and GitHub Copilot (GPT-5) on 50 top GitHub repositories.
- Achieved up to 97% **Attack Success Rate**.
- Agent confidence in the familiar pattern masked the need to run dynamic verification.



Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Bernstein, S., Beste, D., Ayzenshteyn, D., Schonherr, L., & Mirsky, Y. (2026). Trust Me, I Know This Function: Hijacking LLM Static Analysis using Bias. *NDSS'26*

Robust Prompt Failure

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Bernstein, S., Beste, D., Ayzenshteyn, D., Schonherr, L., & Mirsky, Y. (2026). Trust Me, I Know This Function: Hijacking LLM Static Analysis using Bias. *NDSS'26*

Robust Prompt Failure

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Robust Prompt Failure

- We explicitly warned the LLMs about Familiar Pattern Attacks using a detailed system prompt that included an example of a hidden bug.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Robust Prompt Failure

- We explicitly warned the LLMs about Familiar Pattern Attacks using a detailed system prompt that included an example of a hidden bug.

- Results:
  - **Original Prompt:** 16.7% Detection
  - **Robust Prompt:** 16.7% Detection

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Robust Prompt Failure

- We explicitly warned the LLMs about Familiar Pattern Attacks using a detailed system prompt that included an example of a hidden bug.

- Results:
  - **Original Prompt:** 16.7% Detection
  - **Robust Prompt:** 16.7% Detection

Explicit warnings and prompt engineering fail to mitigate the attack, demonstrating that abstraction bias is a deep structural flaw rather than a failure to follow instructions.

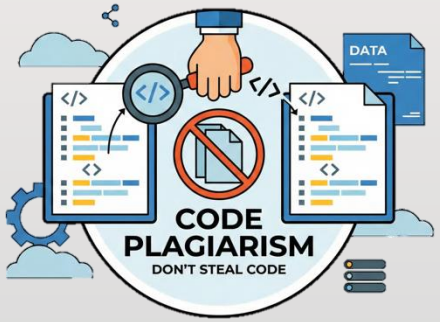
Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Defensive Use

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Defensive Use

Anti-Plagiarism:
Corrupts LLM outputs when attempting to clone or rewrite code.

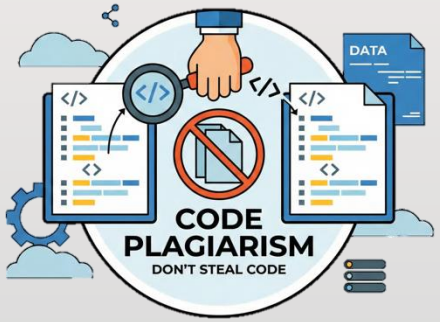


Clean : 84.3%
Infected: 49.7%

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Defensive Use

Anti-Plagiarism:
Corrupts LLM outputs when attempting to clone or rewrite code.



Clean : 84.3%
Infected: 49.7%

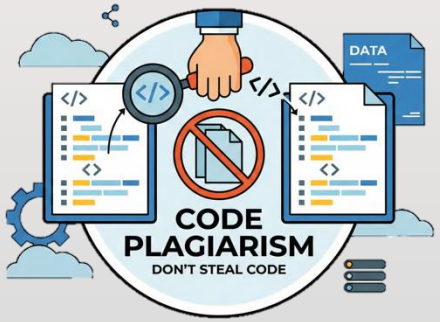
Web Scraping Resistance:
Forces LLM scrapers to extract hidden, irrelevant data.

Clean : 65.5%
Infected: 12.0%

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Defensive Use

Anti-Plagiarism:
Corrupts LLM outputs when attempting to clone or rewrite code.



Clean : 84.3%
Infected: 49.7%

Web Scraping Resistance:
Forces LLM scrapers to extract hidden, irrelevant data.

Clean : 65.5%
Infected: 12.0%

FPAs are inherently dual-use! defenders can weaponize this bias to protect proprietary logic and deter automated scraping.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

What Can We Do?

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

What Can We Do?

Dynamic analysis *can* solve this problem by revealing the true runtime behavior.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

What Can We Do?

Dynamic analysis *can* solve this problem by revealing the true runtime behavior.

However, applying it universally is too expensive, not always possible, slow, and difficult to scale.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

What Can We Do?

Dynamic analysis *can* solve this problem by revealing the true runtime behavior.

However, applying it universally is too expensive, not always possible, slow, and difficult to scale.

Recommendation: Apply dynamic analysis strategically on critical tasks (e.g., vulnerability detection, malware triage) or when handling highly untrusted code.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Conclusions

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Conclusions

Structural Weakness:
Failure is rooted in deep inductive bias, not a failure to follow instructions.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Conclusions

Structural Weakness: Failure is rooted in deep inductive bias, not a failure to follow instructions.

Practical Threat: FPAs are cheap, stealthy, and highly effective against LLMs and code agents

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Conclusions

Structural Weakness: Failure is rooted in deep inductive bias, not a failure to follow instructions.

No Easy Fix: Neither reasoning models nor robust prompts provide a reliable defense.

Practical Threat: FPAs are cheap, stealthy, and highly effective against LLMs and code agents

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Conclusions

Structural Weakness: Failure is rooted in deep inductive bias, not a failure to follow instructions.

Practical Threat: FPAs are cheap, stealthy, and highly effective against LLMs and code agents

No Easy Fix: Neither reasoning models nor robust prompts provide a reliable defense.

Future Goal: We must move toward robust, semantics-aware code understanding to secure LLM agents.

Supported by ERC Starting Grant **AGI-Safety** (GA 101222135)

Thank you!
Questions?

Offensive-ai-lab.com

https://github.com/ShirBernBGU/Trust-Me-I-Know-This-Function

Acknowledgments:

Funding: This work was supported by the European Union (ERC, AGI-Safety, #101222135).

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.