

itp-fuzz: Adversarial Robustness Testing for Interactive Theorem Provers

A framework for finding failure modes in interactive theorem provers (ITPs) and LLM-assisted proof tools.

What This Tool Does (For Programmers)

Imagine you have a super-smart AI assistant that helps you write mathematical proofs. This tool is like a security tester for these AI proof assistants and the underlying proof systems.

Three Main Things It Does:

1.  **Kernel Fuzzing:** Tries to break the core proof checker (the 'kernel')
 - Feeds strange inputs to Lean, Coq, Isabelle, F*
 - Looks for ways to make them accept WRONG proofs (like proving $1=2$)
 - Example: Found that setting `debug.skipKernelTC true` in Lean disables checking
2.  **AI Assistant Testing:** Tricks AI proof helpers with confusing situations
 - Creates proof states designed to fool LLM-based auto completers
 - Checks if they suggest CHEATING tactics like 'sorry' or 'admit'
 - Tests if they recommend dangerous shortcuts that break math
3.  **Living Vulnerability Database:** Tracks known weaknesses
 - Maintains catalog of 21+ known failure modes across ITPs
 - Includes historical bugs, CVEs, and recent discoveries
 - Lets you reproduce known attacks and discover new ones

Quick Verification

```
# Test Coq kernel - should find unsound proof
itp-fuzz kernel --itp coq --iterations 5 --seed 42

# Expected output:
# [UN SOUND] coq-false-via-axiom ← FALSE PROVED!
# [!] Unsoundness detected! Stopping early.
```

Key Files

- `itp-fuzz/harness/coq_interface.py` - Coq proof checker
- `itp-fuzz/kernel_fuzzer/mutants.py` - Attack vectors (including Coq)

- `reports/` - Detailed JSON reports of findings
- `DEMO.md` - Step-by-step usage examples

Status: WORKING

The Coq interface is successfully integrated and detecting kernel soundness bugs!