BACKGROUND

- Bugs in processors present vulnerabilities that are exploitable by well-crafted attacks.
- Verification of security properties can prevent the exploitation of vulnerabilities in a processor.

Overview

- A semi-automated methodology to find security critical invariants (SCI) for use in processor verification.
- A tool chain implementing our methodology.
- An evaluation of SCI Finder on the OR1200 RISC processor.

RESEARCH QUESTION

How to identify the security-critical properties of a processor?

SCI Inference

- We use a penalized logistic regression model with elastic net penalty.
- We manually classify whether an invariant is security-critical or not.
- We model the probability of an invariant i to be security-critical or not as follows (y is the class label):

\[ p_i = \text{probability}(y_i = \text{non security critical}) \]

\[ 1 - p_i = \text{probability}(y_i = \text{security critical}) \]

1. Collecting a set of invariants that govern how processor state is updated.
2. Using published errata, identify those invariants violated by prior, exploitable bugs.
3. Using machine learning, find additional invariants that are critical to security.

RESULTS

1. Properties identified by SCI Finder

<table>
<thead>
<tr>
<th>Properties manually crafted in prior work [1, 2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Example: Link address should not be modified during function call execution


2. Result: Stopping New Bugs.