HyDiff: Hybrid Differential Software Analysis

Yannic Noller  Corina S. Pasareanu  Marcel Böhme
Youcheng Sun  Hoang Lam Nguyen  Lars Grunske
Differential Analysis

Regression Analysis

Side-Channel Analysis

Robustness Analysis of Neural Networks
**Problem**

**Solution**

**Evaluation**

**Summary**

---

**Input**
- Program versions
- Seed input files
- Change-annotated program

---

**Fuzzing**
- Instrumentation
- ICFG
- Import
- Assessment
- Fuzzer output queue
- Mutate inputs

---

**Symbolic Execution**
- Constraint solving / input generation
- Trie extension / assessment
- Exploration
- Symbc output queue

---

**Output**

<table>
<thead>
<tr>
<th>input</th>
<th>+odiff</th>
<th>+ddiff</th>
<th>+crash</th>
<th>+cdiff</th>
<th>+patch-dist</th>
<th>+id</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Set of divergence revealing test inputs:

- **good** in finding shallow bugs, but **bad** in finding deep program paths
- **input reasoning ability**, but **path explosion** and **constraint solving**
HyDiff’s Input

- seed inputs
- **program** under test
- **two different** change types
  1. **inside** the **program code**
  2. **in the input**

input := \textit{change}\textit{(}input_1, input_2\textit{)}

<table>
<thead>
<tr>
<th>Change Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update assignment</td>
<td>(x = x + \text{change}(E_1, E_2);)</td>
</tr>
<tr>
<td>Update condition</td>
<td>if(\text{change}(E_1, E_2)) ...</td>
</tr>
<tr>
<td>Add extra assignment</td>
<td>(x = \text{change}(x, E);)</td>
</tr>
<tr>
<td>Remove assignment</td>
<td>(x = \text{change}(E, x);)</td>
</tr>
<tr>
<td>Add conditional</td>
<td>if(\text{change}(false, C)) ...</td>
</tr>
<tr>
<td>Remove conditional</td>
<td>if(\text{change}(C, false)) ...</td>
</tr>
<tr>
<td>Remove code</td>
<td>if(\text{change}(true, false)) ...</td>
</tr>
<tr>
<td>Add code</td>
<td>if(\text{change}(false, true)) ...</td>
</tr>
</tbody>
</table>

\textbf{change-annotations} by Palikareva et al. [2]


Differential Greybox Fuzzing (DF)

- built upon AFL [1] (genetic algorithm)
- mutant selection driven by differential heuristics:
  - output difference
  - decision history difference
  - cost difference
  - patch distance
- additionally guided by branch coverage

\[\text{Problem}\] 

\[\text{Solution}\] 

\[\text{Evaluation}\] 

\[\text{Summary}\]
Differential Symbolic Execution (DSE)

- built upon Symbolic PathFinder (SPF) [3]
- central data structure: trie
- node selection driven by differential heuristics:
  - decision history difference
  - cost difference
  - patch distance
- additionally guided by branch coverage
HyDiff’s Output

- set of generated inputs
- classified by divergence
  - output difference (+odiff)
  - control-flow (+ddiff)
  - crashing behavior (+crash)
  - execution cost (+cdiff)
- additionally
  - patch distance (+patch-dist)
  - branch coverage (+cov)

Output

<table>
<thead>
<tr>
<th>input</th>
<th>+odiff</th>
<th>+ddiff</th>
<th>+crash</th>
<th>+cdiff</th>
<th>+patch-dist</th>
<th>+cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>id:0001</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>id:0002</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>id:0003</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

set of divergence revealing test inputs
# Experiments

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
<th>Evaluation</th>
<th>Summary</th>
</tr>
</thead>
</table>
| **Regression Analysis** | ▶️ HyDiff classifies all subjects correctly  
▶️ significantly more output and decision differences |  |  |
| **Side-Channel Analysis** | ▶️ HyDiff shows good trade-off between DSE and DF  
▶️ no significant amplification of the exploration |  |  |
| **Robustness Analysis of Neural Networks** | ▶️ stress test for HyDiff  
▶️ HyDiff significantly more output differences |  |  |
HyDiff: Hybrid Differential Software Analysis

Differential Analysis

- Regression Analysis
- Side-Channel Analysis
- Robustness Analysis of Neural Networks

Problem

Solution

Evaluation

Summary

Differential Analysis

- X inputs
- Program P
- Behavior

Input

Output

Differential Symbolic Execution (DSE)

- built upon Symbolic Path Finder (SPF) [3]
- central data structure: trie
- node selection driven by differential heuristics:
  - decision history difference
  - cost difference
  - patch distance
- additionally guided by branch coverage

Side-Channel Analysis

- Robustness Analysis of Neural Networks

Evaluation

- Differential Greybox Fuzzing (DF)
  - built upon AFL [1] (genetic algorithm)
  - mutant selection driven by differential heuristics:
    - output difference
    - decision history difference
    - cost difference
    - patch distance
  - additionally guided by branch coverage

Experiments

- Regression Analysis
  - HyDiff identifies all output differences
  - significantly more output and decision differences
- Side-Channel Analysis
  - HyDiff shows good trade-off between DSE and DF
  - no significant amplification of the exploration
- Robustness Analysis of Neural Networks
  - stress test for HyDiff
  - HyDiff significantly more output differences

Summary

DOI 10.5281/zenodo.3627893

GitHub

yannicnoller/hydiff
References

