Poster: A Tight Integration of Symbolic Execution and Fuzzing

Sébastien Bardin, Yaëlle Vinçont, Michaël Marcozzi

September 15-16, 2022
London, UK
About Fuzzing

• Automatic test input generation

• **Black-box fuzzing** = random generation (usually specification-based)
  
  ![Csmith](image)
  
  2000+ bugs in GCC/LLVM
  1000+ bugs in Z3/CVC4
  400+ bugs in SQL DBMS

• **Grey-box fuzzing** = semi-random generation (code coverage feedback)

  ![SQLancer](image)
  
  Many CVEs and bugs in many apps:
  iOS, Firefox, dpkg, OpenSSH, etc.
Fuzzing vs Symbolic Execution

• Symbolic execution = white-box fuzzing (path constraints solving)

• Grey-box and white-box fuzzing seem complementary
  • Grey-box fuzzing: no path explosion, nor complex path constraints
  • Symbolic execution: easily penetrates paths guarded by an infrequent condition

• Several tools aim at hybrid fuzzing (best of both worlds)
  
  DRILLER  Qsym  Pango/Fin  Angora  MATRYOSHKA  Eclipser
Confuzz: Goals and Principles

• Confuzz = yet another **hybrid fuzzing tool**

• **Goal:** be better than the others 😊

• Based on **two main principles:**
  
  • Rely on grey-box fuzzing to **explore path space**
  
  • For paths beyond **infrequent conditions:**
    
    • Create *easy-enumerable* version of prefix constraint
    
    • Use the grey-box fuzzer to **find correct solutions**
Preliminary Evaluation and Future Work

- **Good preliminary results** vs AFL++ and KLEE, on 3 LAVA-M programs

<table>
<thead>
<tr>
<th></th>
<th>AFL</th>
<th>AFL++</th>
<th>KLEE</th>
<th>Confuzz</th>
</tr>
</thead>
<tbody>
<tr>
<td>base64 Avg</td>
<td>0.0</td>
<td>0.2</td>
<td>10.0</td>
<td>38.8</td>
</tr>
<tr>
<td>3kloc Dev (σ)</td>
<td>0.0</td>
<td>0.4</td>
<td>1.3</td>
<td>0.4</td>
</tr>
<tr>
<td>44 bugs ̂₁₂</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>md5sum Avg</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9</td>
</tr>
<tr>
<td>3kloc Dev (σ)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.7</td>
</tr>
<tr>
<td>57 bugs ̂₁₂</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>uniq Avg</td>
<td>0.0</td>
<td>0.4</td>
<td>5.0</td>
<td>26.9</td>
</tr>
<tr>
<td>3kloc Dev (σ)</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>3.6</td>
</tr>
<tr>
<td>28 bugs ̂₁₂</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>-</td>
</tr>
</tbody>
</table>

- **TODO:** extend constraint language, **evaluate** at scale vs s.o.t.a. tools