Binary Analysis Tools in Cyber Testing

John Merrill and Arch Owen
PROBLEM STATEMENT

• Lack of source code in cyber engagements drives need for binary analysis tools, however...

• Binary-only analysis tools are limited and less developed than source analysis tools

• Draper has implemented **Modular, Open** software frameworks for static and dynamic analysis of **binary** code:
  - Dynamic analysis – VADER
  - Static analysis – SHREDDER.

• These frameworks allow for shared development and use by a broad contractor community.

• Draper has leveraged the inherent modularity of these frameworks to implement new advanced binary analysis capabilities.
### Survey of Example Binary Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Static Analysis</th>
<th>Dynamic Analysis</th>
<th>Usability/Adaptability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Automated Static Analysis</td>
<td>Reverse Engineering</td>
<td>Fuzzing</td>
</tr>
<tr>
<td>CodeSonar</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IDA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ghidra</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Binary Ninja</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AFL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Angr</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mayhem</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Binary analysis tools exist BUT
- Few are open AND modular
- None are designed for joint use
QUALITATIVE COMPARISON OF BINARY TOOLS

We need a means to efficiently leverage all techniques AND facilitate more rapid binary tool development and maturation.

Ease of Deployment | Memory Corruption Bugs | Logic Bugs
---|---|---
Automated Static Analysis | Reverse Engineering | Fuzzing | Symbolic / Concolic Execution

Potentially very effective but VERY manpower intensive and dependent

Very easy to employ, but lower performance than other options

A balance between ease-of-deployment and performance
Draper’s Approach for Rapid Tool Development

• Rapid, robust tool development and maturation requires a 2-step approach

• Step 1: Develop a **modular, open, automated** framework
  – Automation: easy deployment
  – Modular and Open: easy development of new features

• Step 2: Rapidly develop and integrate novel techniques
VADER Dynamic Analysis
Traditional software testing handles known edge cases. Fuzzing, in contrast, attempts to expose unknown edge cases through randomly generated tests.
Fuzzers have tended to focus on source code, and the monolithic nature of available options limits innovation and advanced features.
VADER open modularity leverages the best features of different fuzzers AND facilitates binary support, more rapid adaptation, and faster tool development.
VADER CURRENT FEATURES

• Modular support for many existing open-source fuzzers
  – AFL
  – Radamsa
  – Domato

• Automated sample generation and crash triage

• Draper developed capabilities leveraging open modularity:
  – Custom Black Box **Taint Tracking** subsystem
    ◦ Ability to traverse complex code paths
    ◦ Full support for binary code
  – Real time, automated fuzzer **switching** - faster AND deeper code search
Reverse engineering is a time-consuming task involving numerous steps of information gathering. Many of these steps can be automated; however, the means to do so are limited.
The SHREDDER framework automates a large collection of common reverse engineering tasks and provides a simple interface for user extensions.
SHREDDER CURRENT FEATURES

• Integration of numerous static analysis jobs

• IDA / Binary Ninja plugin support

• Draper developed capabilities leveraging open modularity:
  – Library Identification capability
    ◦ Precise signature detection for library versions in binary
    ◦ Correlation with CVE database
**Problem Statement**

- Lack of source code in cyber engagements drives need for binary analysis tools, however...

- Binary-only analysis tools are limited and less developed than source analysis tools

- Draper has implemented **Modular, Open** software frameworks for static and dynamic analysis of **binary** code:
  - Dynamic analysis – VADER
  - Static analysis – SHREDDER.

- These frameworks allow for shared development and use by a broad contractor community.

- Draper has leveraged the inherent modularity of these frameworks to implement new advanced binary analysis capabilities.