SCALABLE HUMAN-COMPETITIVE SOFTWARE REPAIR

Stephanie Forrest
Michael Dewey-Vogt
Claire Le Goues
Westley Weimer

http://genprog.cs.virginia.edu
“Everyday, almost 300 bugs appear […] far too many for only the Mozilla programmers to handle.”

– Mozilla Developer, 2005

PROBLEM: BUGGY SOFTWARE

Average time to fix a security-critical error: 28 days.

Annual cost of software errors in the US: $59.5 billion (0.6% of GDP).

10%: Everything Else
90%: Maintenance
BUG BOUNTIES: $20-$3000+ PER PATCH

Bug Bounty Program

Introduction

The Mozilla Security Bug Bounty Program is designed to encourage security research in Mozilla software and to reward those who help us create the safest Internet clients in existence.

Many thanks to Linspire and Mark Shuttleworth, who provided start-up funding for this endeavor.

General Bounty Guidelines

Encouraging More Chromium Security Research

Thursday, January 28, 2010

Labels: google/chrome, security

In designing Chromium, we’ve been working hard to make the browser as secure as possible. We’ve made strong improvements with the integrated sandboxing and our up-to-date user base. We’re always looking to stay on top of the latest browser security features. We’ve also worked closely with the broader security community to get independent scrutiny and to quickly fix bugs that have been reported.

Some of the most interesting security bugs we’ve fixed have been reported by researchers external to the Chromium project. For example, the same origin policy bypass by Isaac Dawson or this v8 engine bug found by the Mozilla Security Team. Thanks to the collaborative efforts of these people and others, Chromium security is stronger and our users are safer.

Today, we are introducing an experimental new incentive for external researchers to participate. We will be rewarding select interesting and original vulnerabilities reported to us by the security research community. For existing contributors to Chromium security—who would likely continue to contribute regardless — this may be seen as a token of our appreciation. In addition, we are hoping that the introduction of this program will encourage new individuals to participate in Chromium security. The more people involved in scrutinizing Chromium’s code and behavior, the more secure our millions of users will be.

Such a concept is not new; we’d like to give serious kudos to the folks at Mozilla for their long-running and impecable vulnerability reward program.

Any valid security bug filed through the Chromium bug tracker (under the template “Security Bug”) will qualify for consideration. As this is an experimental program, here are some guidelines in the form of questions and answers.

Q: What reward might I get?

A) As per Mozilla, our base reward for eligible bugs is $500. If the panel finds a particular bug noticeably severe or notoriouse, closer to a maximum rewards of $3000. The panel may also decline.

<table>
<thead>
<tr>
<th>Tarsnap Bug Bounties</th>
</tr>
</thead>
<tbody>
<tr>
<td>According to <a href="http://en.wikipedia.org/wiki/Linus%27_Law">Linus' Law</a>, &quot;given enough eyeballs, all bugs are bound to be found.&quot; This is one of the reasons why the Tarsnap client source code is available; but merely making the source code available doesn't solve anything if people don't bother to read it.</td>
</tr>
</tbody>
</table>

For this reason, Tarsnap has a series of **bug bounties**. Some bug bounties offered by [Mozilla](http://www.mozilla.org) and [Google](http://www.google.com), the Tarsnap bug bounty offers an opportunity for people who find bugs to win cash. Unlike these other bounty programs, Tarsnap bug bounties aren't limited to security bugs.
E.G., GOOGLE PAID $11,500 IN BOUNTIES BETWEEN MAY 23, 2012 AND JUN 26, 2012

Tarsnap:
- 125 spelling/style
- 63 harmless
- 11 minor
- 1 major

$75/200 = 38\%$ TP rate

$\$17 + 40$ hours per TP

which were wrong yet didn't actually affect the compiled code.

But most importantly, $1265$ of bugs gives me the peace of mind of knowing that I'm not the only person who has looked at the Tarsnap code, and if there are more critical bugs like the security bug I fixed in January, they've escaped more than just my eyeballs. Worth the money? Every penny.
GENPROG: EVOLVING SOFTWARE REPAIRS

INPUT

EVALUATE FITNESS

DISCARD

ACCEPT

MUTATE

OUTPUT

http://genprog.cs.virginia.edu
WHY WE ARE HUMAN COMPETITIVE

Effective:
- Tested on 105 human-repaired bugs in over 5 million LOC
- GenProg automatically repaired 60 (57%)
- Tarsnap CEO found 38% rate “worth every penny”
- Security repairs tested using Microsoft’s fuzz-testing std

Cheap: $7.32 per TP (successful bug fix)
- Tarsnap paid $17 per TP, IBM pays $25

Fast: 96 minutes (wall clock)
- Compared to 40 hours for Tarsnap

Quality (ISSTA to appear):
- GenProg-patched code + machine-generated documentation
  is more maintainable than
- Human-generated patches + commit message
Question: “If I were to use your technique on the next 100 bugs that were filed against my project, how many would it fix, how much would that cost, and how long would it take?”

Goal: a large set of important, reproducible bugs in non-trivial programs.

Approach: use historical data of important, reproducible bugs in non-trivial programs

- Consider popular programs from SourceForge, Google Code, Fedora SRPM, etc
- Bugs merited a developer-written test case and a bug report “severity” of 3/5 or more
- Use all pairs of viable versions from source control repositories.
- “Lock in” our algorithm first, then gather up all bugs.
- Evaluate in Amazon EC2 cloud
## Benchmarks

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
<th>LOC</th>
<th>Tests</th>
<th>Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbc</td>
<td>Language (legacy)</td>
<td>97K</td>
<td>773</td>
<td>1</td>
</tr>
<tr>
<td>gmp</td>
<td>Multiple precision math</td>
<td>145K</td>
<td>146</td>
<td>1</td>
</tr>
<tr>
<td>gzip</td>
<td>Data compression</td>
<td>491K</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>libtiff</td>
<td>Image manipulation</td>
<td>77K</td>
<td>78</td>
<td>17</td>
</tr>
<tr>
<td>lighttpd</td>
<td>Web server</td>
<td>62K</td>
<td>295</td>
<td>5</td>
</tr>
<tr>
<td>php</td>
<td>Language (web)</td>
<td>1,046K</td>
<td>8,471</td>
<td>31</td>
</tr>
<tr>
<td>python</td>
<td>Language (general)</td>
<td>407K</td>
<td>355</td>
<td>1</td>
</tr>
<tr>
<td>wireshark</td>
<td>Network packet analyzer</td>
<td>2,814K</td>
<td>63</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>5,14M</td>
<td>10,193</td>
<td>60</td>
</tr>
</tbody>
</table>

http://genprog.cs.virginia.edu
In 2009, we demonstrated that it was possible to repair bugs using GP

- Evaluated on small/toy programs with small test suites, no direct cost comparisons, no systematic quality comparisons

2012: human-competitive scalable repairs for off-the-shelf, real-world bugs

- ~100x more code, ~200x more tests, ~10x more bugs (and bugs that matter!), systematic study, direct time measurements (e.g., 96 minutes vs. 40 hours), direct cost measurements (e.g., $8 vs. $17), direct maintainability measurements
GenProg addresses a critical and challenging problem (0.6% US GDP)

Better than humans on quantitative metrics used in software industry.

Systematic selection of benchmark programs and bugs

Scalability achieved through algorithmic innovations