

SCALABLE HUMAN-COMPETITIVE SOFTWARE REPAIR



**Stephanie
Forrest**



**Michael
Dewey-Vogt**



**Claire
Le Goues**



**Westley
Weimer**



“Everyday, almost 300 bugs appear [...] far too many for only the Mozilla programmers to handle.”



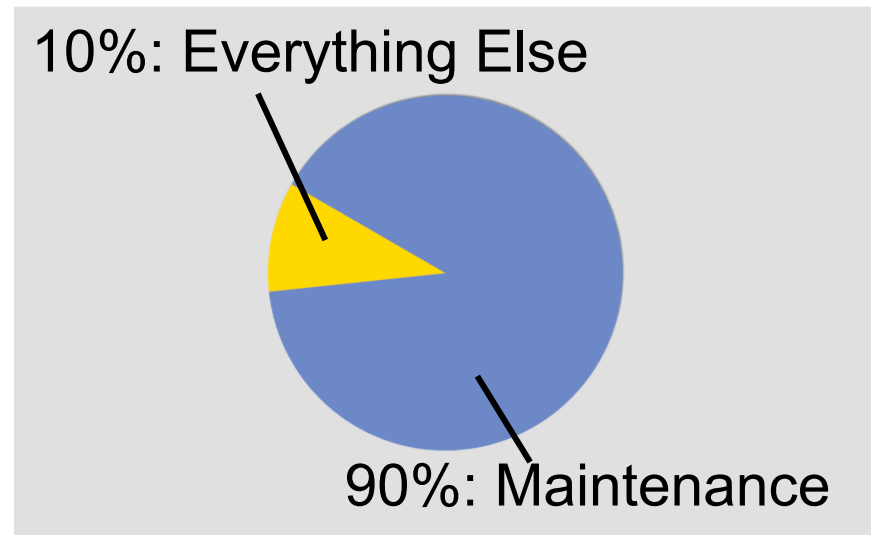
mozilla

– *Mozilla Developer,*
2005

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

PROBLEM: BUGGY SOFTWARE

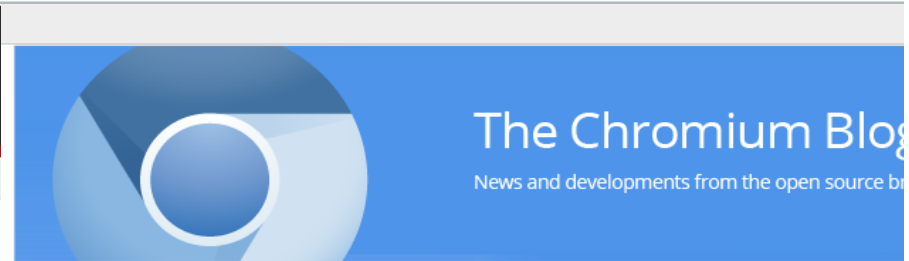
Average time to fix a security-critical error: 28 days.



BUG BOUNTIES: \$20-\$3000+ PER PATCH



The screenshot shows the Mozilla website's navigation bar with the Mozilla logo and links for 'About Us', 'Community Map', 'Our Projects', and 'Get...'. Below the navigation is a large red heading 'Bug Bounty Program' followed by an 'Introduction' section. The introduction text states: '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.' Below the introduction is a section titled 'General Bounty Guidelines'.



The screenshot shows the header of the Chromium Blog. It features the Chromium logo on the left and the text 'The Chromium Blog' on the right, with a subtitle 'News and developments from the open source browser' below it.

Encouraging More Chromium Security Research

Thursday, January 28, 2010

Labels: [googlechrome](#), [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, [this same origin policy bypass from 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 successful 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 particularly severe or particularly clever we envisage rewards of \$1337. The panel may also decide

Tarsnap

Online backups for the truly paranoid

- Tarsnap
- News
- About
- Legal
- Infrastructure
- Bug Bounty
- Winners
- Design

Tarsnap Bug Bounties

According to [Linus' Law](#), "given enough eyeballs, all bugs are shallow." This is one of the reasons why the Tarsnap client source code is available; but merely making the source code available doesn't do anything if people don't bother to read it.

For this reason, Tarsnap has a series of *bug bounties*. Similar bounties offered by [Mozilla](#) and [Google](#), the Tarsnap bug bounty offers an opportunity for people who find bugs to win cash. Unlike those offered by Mozilla and Google, the Tarsnap bug bounties aren't limited to security bugs. Depen

Search

Archive
April

Subscribe
RSS



More
Visit
inform

Useful
[Chro](#)
[Gooc](#)
[Gooc](#)
[Gooc](#)
[Gooc](#)
[Gooc](#)

E.G., GOOGLE PAID \$11,500 IN BOUNTIES BETWEEN MAY 23, 2012 AND JUN 26, 2012

Home > Security

News

Google calls, raises Mozilla's bug bounty for Chrome flaws

Boosts cash-for-bugs maximum payment to \$3,133, makes researchers mostly happy

By [Gregg Keizer](#)
July 22, 2010 11:59 AM ET

2 Comments [+ Briefcase](#) [What's this?](#)

Computerworld - Google on Tuesday hiked bounty payments for Chrome bugs to a maximum of \$3,133, up almost \$2,000 from the previous top dollar payout of \$1,337.

The move came less than a week after rival browser maker [Mozilla increased Firefox bug bounties](#) to \$3,000.

In an entry to the [Chromium project's blog](#), Chris Evans, who works on the Chrome security team, announced the new maximum bounty of \$3,133.70 and said Google would "most likely" award that amount for all vulnerabilities rated "critical" in the company's four-step scoring system.

"The increased reward reflects the fact that the sandbox makes it harder to find bugs of this severity," said Evans, referring to the technology baked into Chrome that isolates processes from one another and the rest of the machine, preventing or at least hindering malicious code from escaping an application to wreak havoc or infect the computer.

Tarsnap:

125 spelling/style
63 harmless
11 minor
+ 1 major

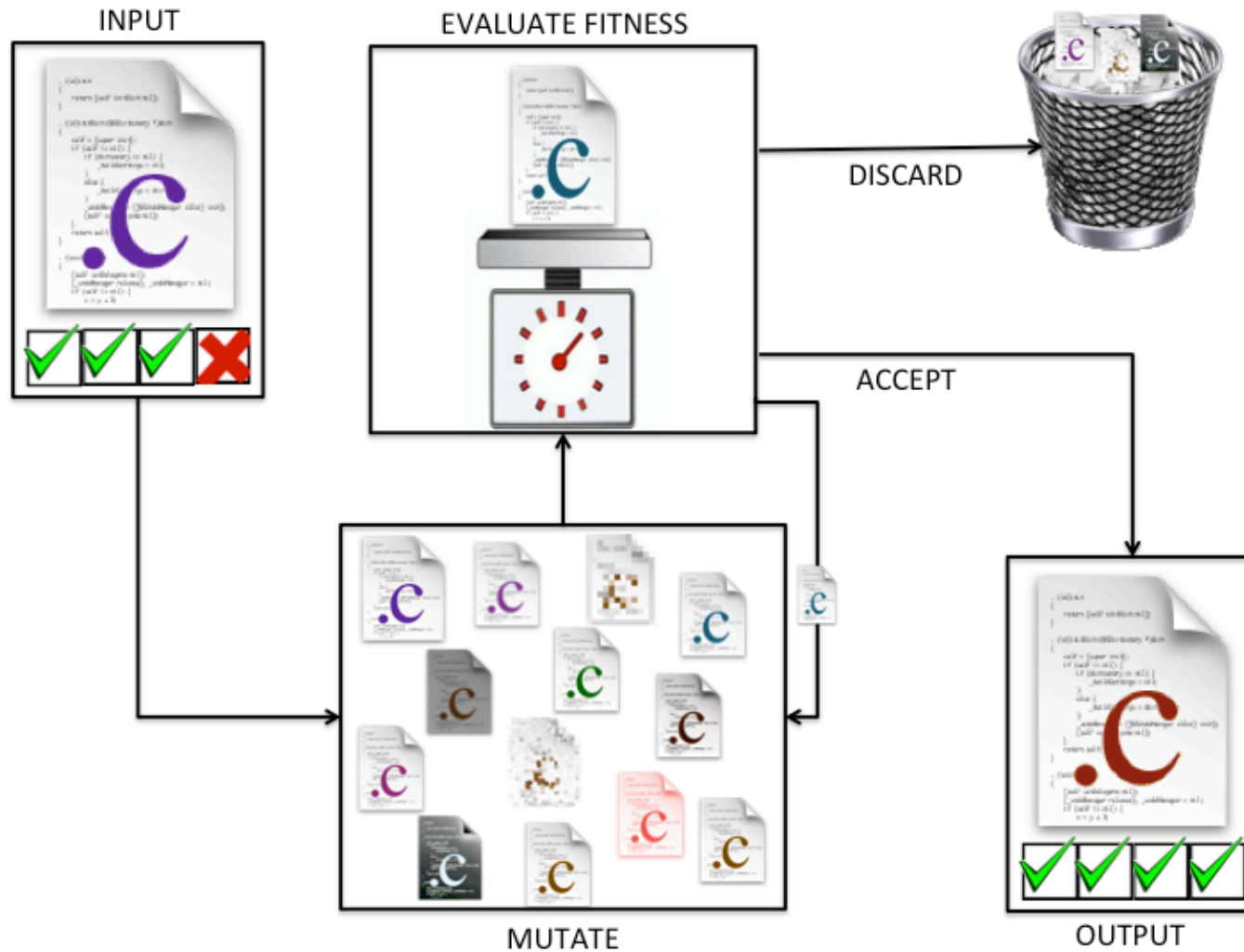
75/200 = 38% TP rate

\$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



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

SYSTEMATIC EVALUATION

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

Program	Description	LOC	Tests	Bugs	
				Fixed	Total
fbc	Language (legacy)	97K	773	1	3
gmp	Multiple precision math	145K	146	1	2
gzip	Data compression	491K	12	1	5
libtiff	Image manipulation	77K	78	17	24
lighttpd	Web server	62K	295	5	9
php	Language (web)	1,046K	8,471	31	44
python	Language (general)	407K	355	1	11
wireshark	Network packet analyzer	2,814K	63	3	7
Total		5,14M	10,193	60	105

SCALABILITY

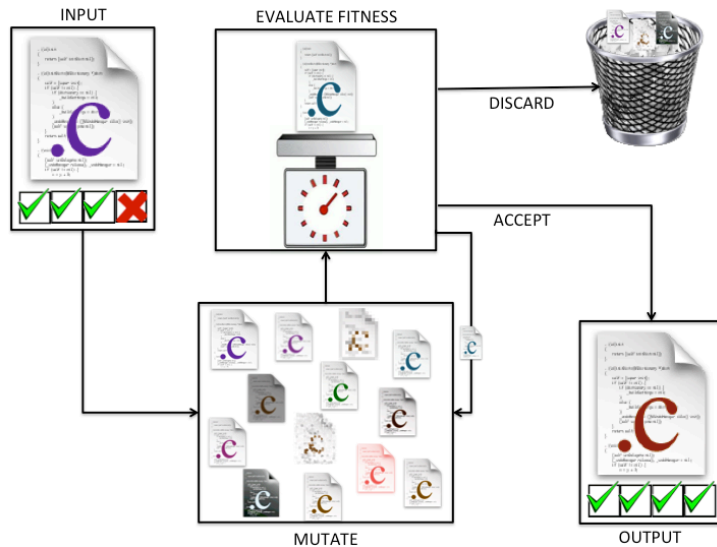
In 2009, we demonstrated that it was **possible** to repair bugs using GP

- 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

CONCLUSION



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