


“Every day
almost 300 bugs appear [...]
far too many for only the
Mozilla programmers to
handle [1]”



“The US economy loses \$60 billion each year in costs associated with [...] distributing software patches [2].”

“Average time to fix a security-critical error: 28 days. [3]”

“The US economy loses \$60 billion
each year in costs associated with [...]
distributing software patches [2].”

PROBLEM: BUGGY SOFTWARE

“Average time to fix a security-critical
error: 28 days. [3]”

HOW BAD IS IT?

Mozilla reserves the right to not give a bounty payment if we believe the actions of the reporter have endangered the security of Mozilla's end users.

If two or more people report the bug together the reward will be divided among them.

Client Reward Guidelines

The bounty for valid critical client security bugs will be \$3000 (US) cash reward and a Mozilla T-shirt. The bounty will be awarded for [sg:critical](#) and [sg:high](#) severity security bugs that meet the following criteria:

- Security bug is present in the most recent supported, beta or release candidate version of Firefox, Thunderbird, Firefox Mobile, or in Mozilla services which could compromise users of those products, as released by Mozilla Corporation or Mozilla Messaging.
- Security bugs in or caused by additional 3rd-party software (e.g. plugins, extensions) are excluded from the Bug Bounty program.

More information about this program can be found in the [Client Security Bug Bounty Program FAQ](#).

Web Application and Services Reward Guidelines

The bounty for valid web applications or services related security bugs, we are giving a range starting at [\\$500](#) (US) for high severity and, in some cases, may pay up to [\\$3000](#) (US) for extraordinary or critical vulnerabilities. We will also include a Mozilla T-shirt. The bounty will be awarded for [ws:critical](#) and [ws:high](#) security bugs that meet the following criteria:

- Security bug is present in the web properties outlined in the [Web Application Security Bounty FAQ](#)
- Security bug is on the list of sites which part of the bounty. See the [eligible bugs](#) section of the [Web Application Security Bounty FAQ](#) for the list of sites which is included under the bounty.

More information about this program can be found in the [Web Application Security Bounty FAQ](#).

an opportunity for people who find bugs to win cash. Unlike those bounties, the Tarsnap bug bounties aren't limited to security bugs. Depending on the type of bug and when it is reported, different bounties will be awarded:

Bounty value	Pre-release bounty value	Type of bug
\$1000	\$2000	A bug which allows someone intercepting Tarsnap traffic to decrypt Tarsnap users' data.
\$500	\$1000	A bug which allows the Tarsnap service to decrypt Tarsnap users' data.
\$500	\$1000	A bug which causes data corruption or loss.
\$100	\$200	A bug which causes Tarsnap to crash (without corrupting data or losing any data other than an archive currently being written).
\$50	\$100	Any other non-harmless bugs in Tarsnap.
\$20	\$40	Build breakage on a platform where a previous Tarsnap release worked.
\$10	\$20	"Harmless" bugs, e.g., cosmetic errors in Tarsnap output or mistakes in source code comments.
\$1	\$2	Cosmetic errors in the Tarsnap source code or website, e.g., typos in website text or source code comments. Style errors in Tarsnap code qualify here, but usually not style errors in upstream code (e.g., libarchive).

The pre-release bounty value will be awarded for bugs reported in the interval between when a new Tarsnap release is sent to the [tarsnap](#)

**SOLUTION:
PAY STRANGERS**

SOLUTION:

~~PAY STRANGERS~~

**SOLUTION:
AUTOMATE!!!**

Automated Patching Using Genetic Programming

TEAM 8

Peerapon Akkapusit, Zuzana Jelčicová,
Zelalem Mihret, Peter Muschick

Related work

“Automatically Finding Patches Using Genetic Programming”

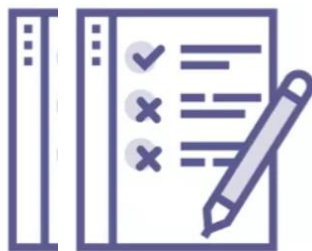
Westley Weimer, Thanh Vu Nguyen, Claire Le Goues, Stephanie Forrest

“A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for \$8 Each”

Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer



Buggy program

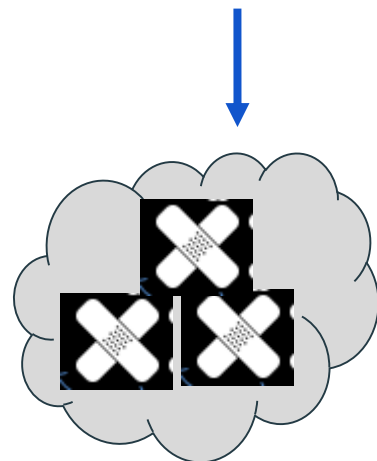


Fault localization

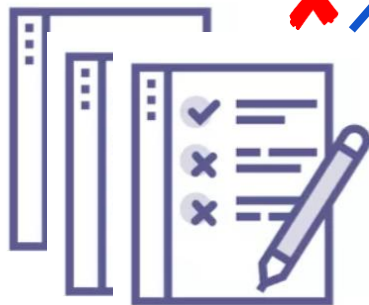


**mutation
crossover**

GA operations



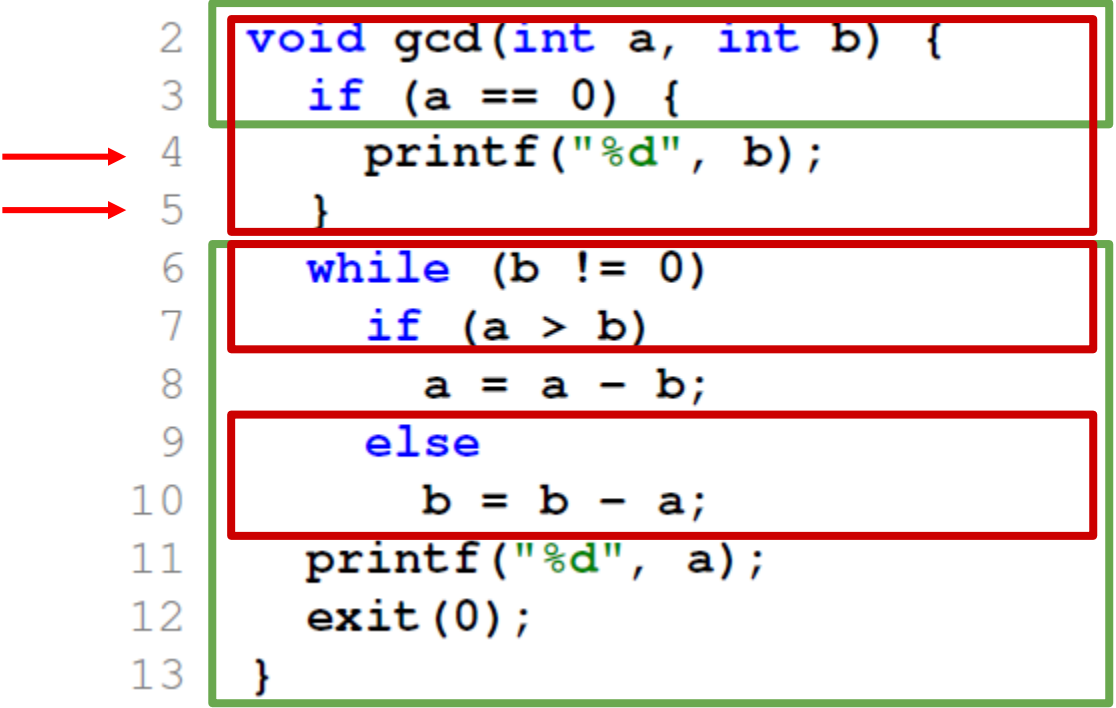
Candidate patch space



Test suite (Fitness function)



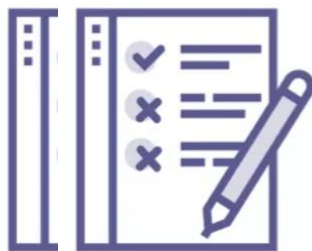
```
1  /* requires: a >= 0, b >= 0 */
2  void gcd(int a, int b) {
3      if (a == 0) {
4          printf("%d", b);
5      }
6      while (b != 0)
7          if (a > b)
8              a = a - b;
9          else
10             b = b - a;
11     printf("%d", a);
12     exit(0);
13 }
```



The diagram illustrates the execution flow of the gcd function. A green box encloses the entire function body from line 2 to line 13. Red boxes highlight specific code blocks: the initial if-statement (lines 3-5), the while-loop's if-statement (lines 7-9), and the else block (lines 9-10). Two red arrows point to lines 4 and 5, indicating the execution path when a == 0.



Buggy program

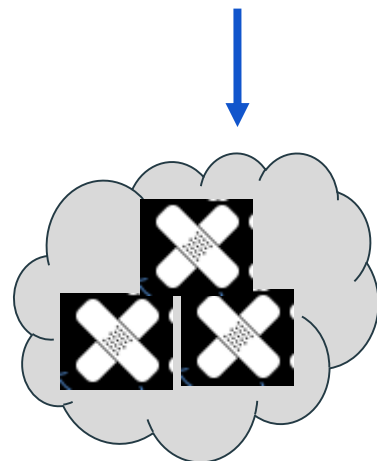


Fault localization

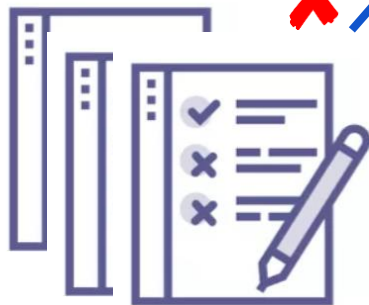


**mutation
crossover**

GA operations



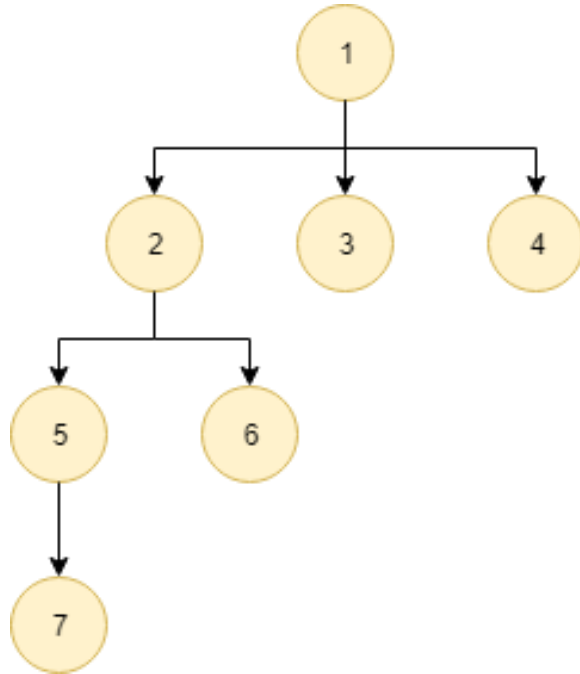
Candidate patch space



Test suite (Fitness function)

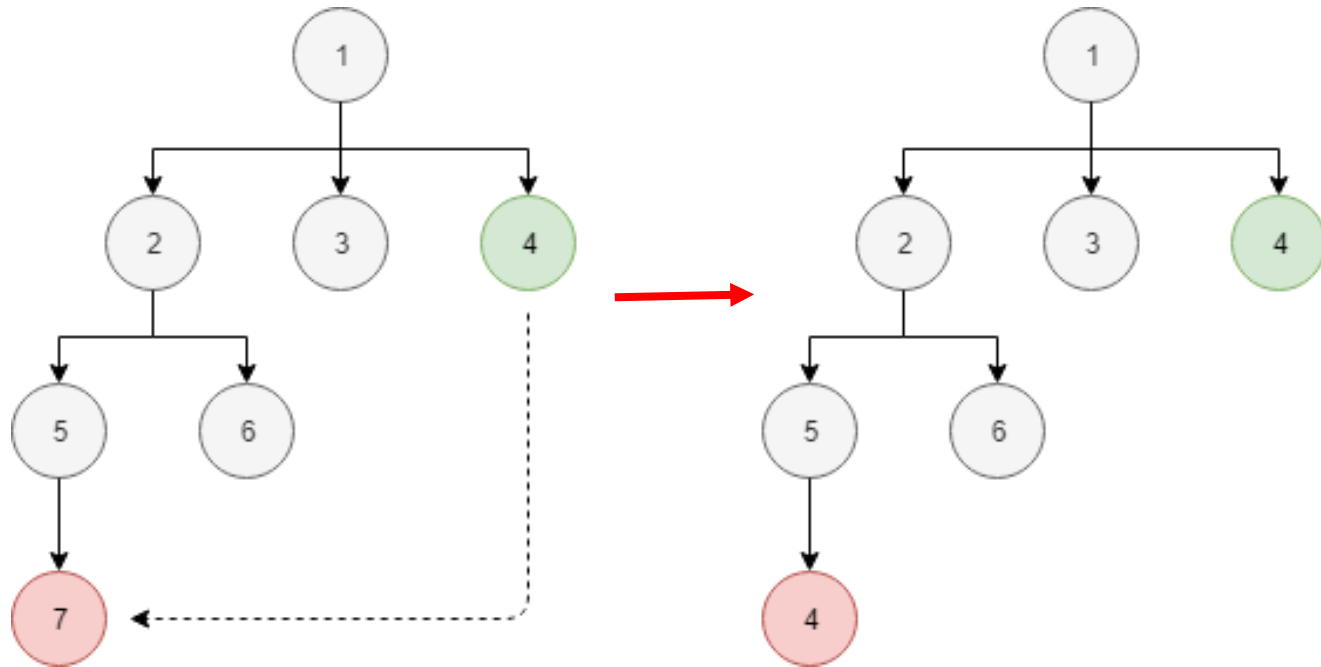


Mutations

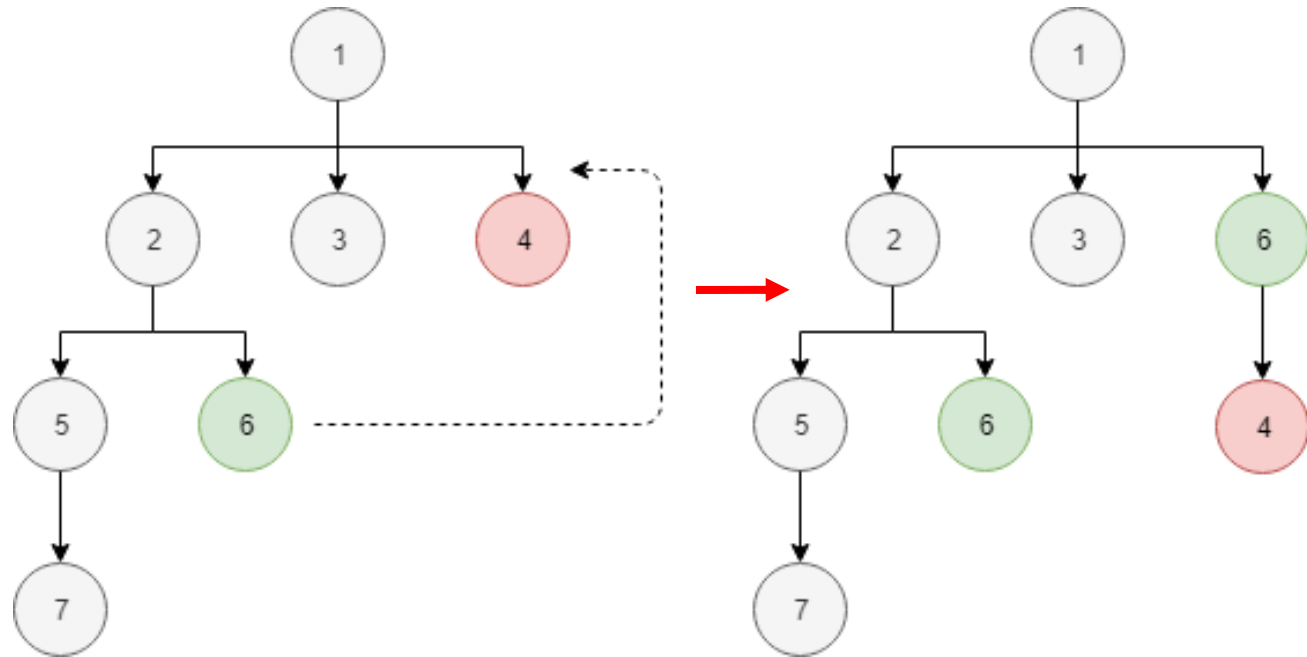


- Replace
- Insert
- Delete

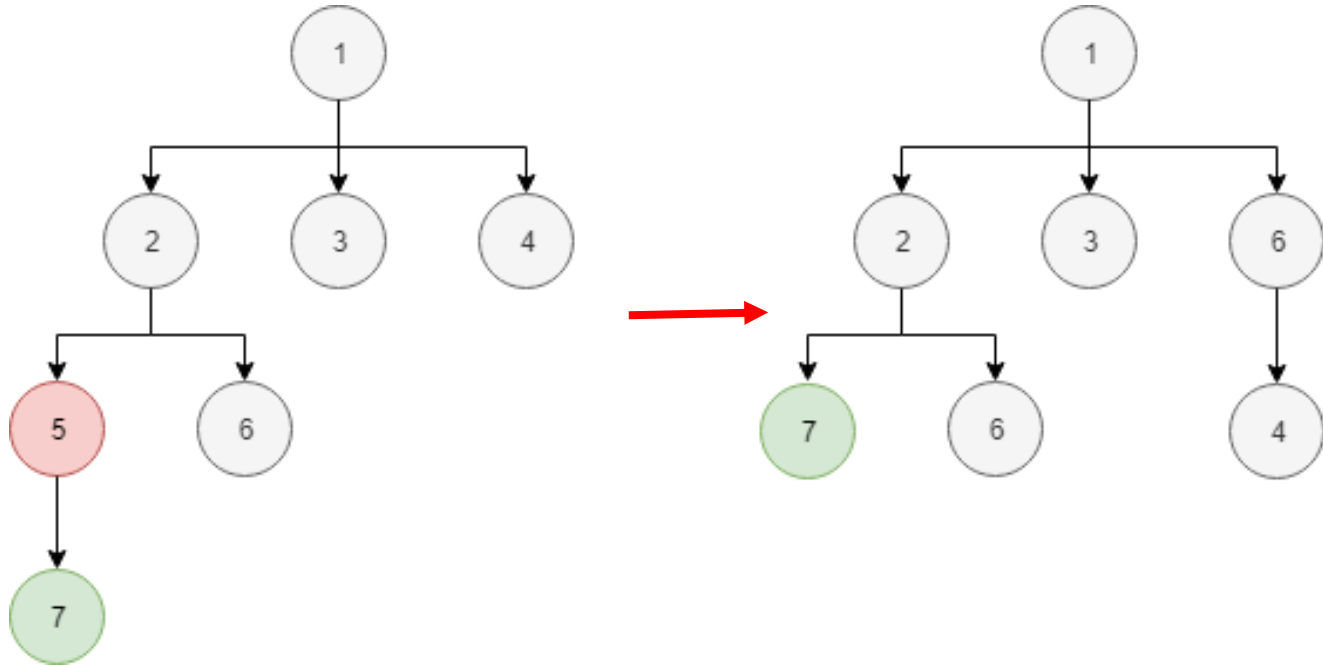
Replace



Insert

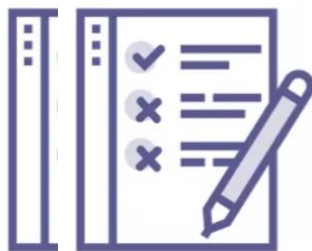


Delete





Buggy program

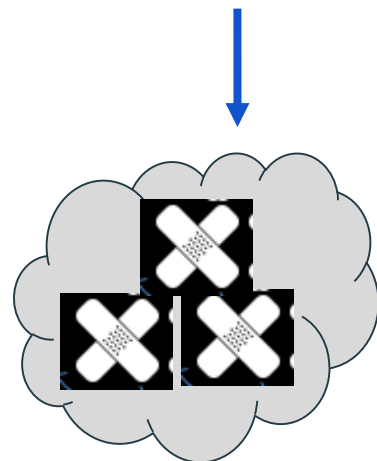


Fault localization

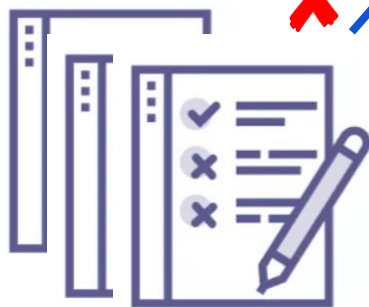


**mutation
crossover**

GA operations



Candidate patch space



Test suite (Fitness function)



Sources

- [1] John Anvik, Lyndon Hiew and Gail C. Murphy. “Who Should Fix This Bug?” University of British Columbia, 2006
- [2] Zhivich, Michael, and Robert K. Cunningham. “The Real Cost of Software Errors.” IEEE Security & Privacy Magazine 7.2 (2009)
- [3] Symantec. “Symantec Internet Security Threat Report Trends for January 06–June 06”. 2006
- [4] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, Westley Weimer. “A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for \$8 Each”

THANK YOU!
Q/A