



code intelligence

Breakthrough Dynamic Application Testing

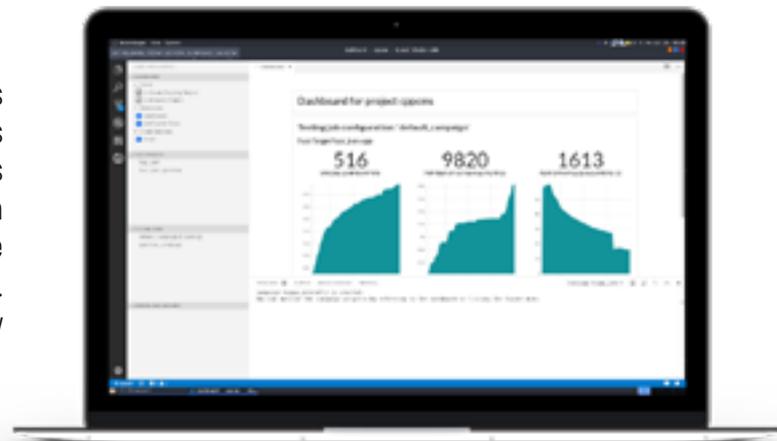
Code Intelligence enables you to simplify your software testing processes. Our solution - CI Fuzz - tests your code for bugs and critical vulnerabilities with the help of modern fuzzing technologies. It moves the testing process into the agile development process and empowers developers to be the first line of defense in your company by enabling them to test their own code. We provide you with a simple, scalable and automated testing solution to help you make your code quality as outstanding as your applications.

What is feedback-based fuzzing?

Fuzzing is fantastic at finding edge cases. It is able to generate and execute millions of tests and is the best way to find bugs hidden deeply in the code. Traditional fuzzing was based on generating random inputs. By contrast, feedback-based fuzzing uses instrumentation to generate inputs based on the application feedback, e.g. whether the last input reached a new program state or what kind of instructions have been applied to the input. The main goal here is to maximize code coverage and thus triggering more bugs in the software under test.

Introducing CI Fuzz

We help you to solve the problems regarding continuous testing for security and reliability. Our approach combines the advantages of static and dynamic software analysis methods. By using coverage-based fuzzing in combination with symbolic code execution and sanitizers, it is possible to monitor the behavior of your program during runtime. This allows us to create more valid inputs triggering new program states, but also detecting subtle bugs.



IDE Plugin

Test and review code without leaving your favorite IDE. In contrast to expert-level fuzzers such as AFL, CI Fuzz integrates directly into the IDE of developers, allowing them to set up and improve automated fuzz tests in an interactive and feedback-driven manner.

- See which parts of the code were already reached
- Supply additional input grammars for structure-aware fuzzing
- Browse the issues found by the fuzz tests

Continuous Integration

CI Fuzz integrates seamlessly into your workflow and continuously tests your software for security and reliability. Once the automated fuzz tests are written, our fully automated Continuous Integration component ensures that the fuzz tests are executed with every new change and incidents are reported timely.

- Integrate fuzzing into a standard workflow
- Perform tests on each code change
- Fuzz on a Kubernetes cluster



code intelligence

About us

Code Intelligence is a software security company headquartered in Bonn, Germany. Our vision is to empower every company to test their own code by providing an outstanding testing solution built for developers and testing experts alike. Our customers include financial institutions, telecommunication providers and global technology companies.



Benefits at a glance

Company

Gain security & reliability

The security and reliability of your software products are two of your organization's most valuable assets.

We help you to provide these assets with our modern security testing solution.

Management

Reduce costs

Integrate IT security know-how into your company with less need for expensive professionals.

Avoid legal costs, fines and penalties due to fewer security issues.

Developers/(Pen)testers

Save time

Accelerate your software development process and save valuable time by applying our security test suite.

Run tests faster and more efficiently without needing to filter false positives manually.

Developers/(Pen)testers

Optimize processes

Most security software testing processes are release-driven and don't fit into the agile software development process.

Optimize your software development process by integrating continuous quality management.

Code Intelligence GmbH

Rheinwerkallee 6

53227 Bonn

www.code-intelligence.com

Contact

Jonathan Reimer

+49 228 2869 5830

sales@code-intelligence.com

Follow us

 @CI_GmbH

 [company/codeintelligence/](https://www.linkedin.com/company/codeintelligence/)