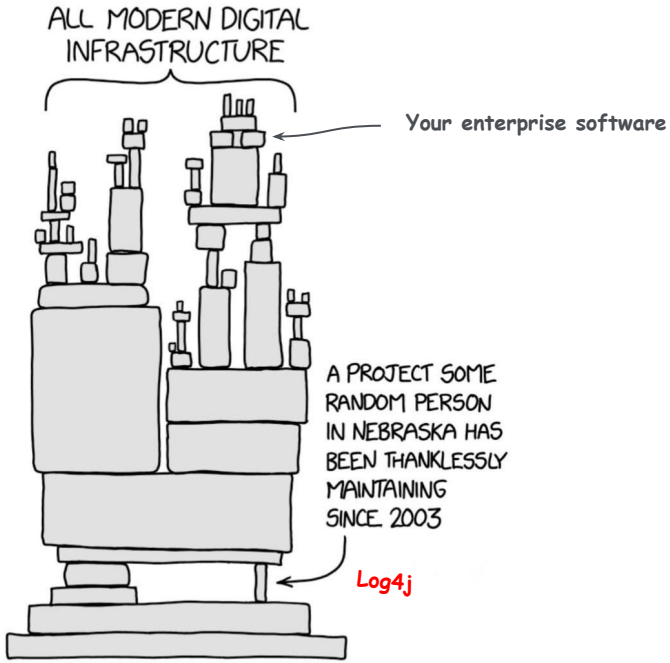
code intelligence

Lessons from Log4j

Continuous Security Testing of open-source components is possible!

OSS can be a security risk in your supply chain

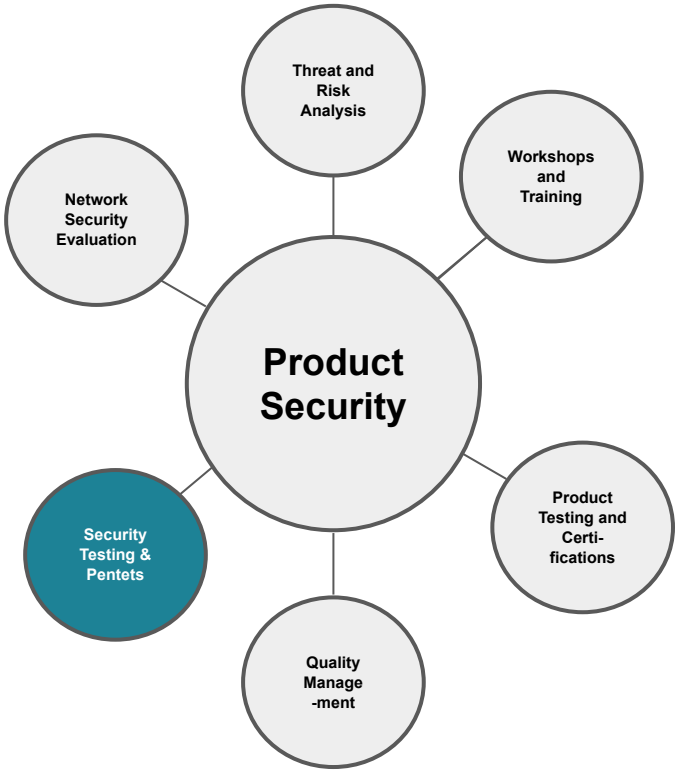
code intelligence



Aspects of product security

code intelligence

“There are no shortcuts!
All open-source components
in your software supply chain
need to be tested”



What can we learn from large tech companies like Google or Alibaba?

code intelligence

Alibaba fixed more than 100 critical vulnerabilities[1] in **fastjson2**, since 2021

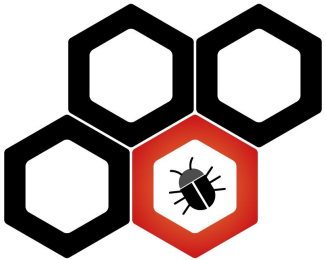
- The library gets continuously tested
- With state-of-the-art security tests
- And they use this service for free

But How?

Fastjson2 is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Fastjson can work with arbitrary Java objects, including pre-existing objects that you do not have source-code of.

[1]https://bugs.chromium.org/p/oss-fuzz/issues/list?q=alibaba%20type%3DBug-Security&can=1

Their secret weapon: OSS-Fuzz



A service to secure open-source software at scale

As of July 2022, OSS-Fuzz has found
over **40,500** bugs in **650** open source projects.

Which projects qualify for OSS-Fuzz?

code intelligence

master · oss-fuzz / projects / · Go to file

ksztyber spdk: update Michal Berger's e-mail (#8591) ··· ff4a2d0 4 hours ago History

..		
abseil-cpp	abseil-cpp: fix build (#8015)	2 months ago
adal	Add vendors to Python projects. (#8547)	15 hours ago
aiohttp	Add vendors to Python projects. (#8547)	15 hours ago
airflow	Add vendors to Python projects. (#8547)	15 hours ago
alembic	Set flags to use old pass manger (#7828)	4 months ago
ansible	Add vendors to Python projects. (#8547)	15 hours ago
antlr4-java	Initial commit [antlr4] (#8019)	2 months ago
apache-commons-cli	[apache-commons-cli] Initial Integration (#8255)	last month
apache-commons-codec	Update project.yaml for several projects (#8032)	2 months ago
apache-commons-collections	[commons-collections] Initial Integration (#8280)	last month
apache-commons-configuration	Changes to apache-commons-configuration and new fuzz targets for http...	20 days ago
apache-commons-io	adding project maintainer (#8386)	25 days ago
apache-commons-jxpath	Update project.yaml for several projects (#8032)	2 months ago
apache-commons-lang	Update project.yaml for several projects (#8032)	2 months ago
apache-commons-logging	[apache-commons-logging] Initial Integration (#8362)	28 days ago
apache-commons	[apache-commons] Add some fuzz-targets for commons-math (#8089)	2 months ago

- ☑ Is it open-source?
- ☑ Is it relevant for a broader community?
- ☑ Is it actively maintained?
- ☑ Are the maintainers willing to cooperate?
- ☑ Does it meet technical prerequisites?

https://github.com/google/oss-fuzz/tree/master/projects

Do your open-source components qualify for OSS-Fuzz?

code intelligence

Probably, yes!

If you're not sure, please reach out to our open-source security team
oss-security@code-intelligence.com

Khaled Yakdan

Co-founder & Chief Scientist

Norbert Schneider

Open-Source Security Engineer

Fabian Meumertzheim

Open-Source Security Engineer

How to onboard new open-source projects to OSS-Fuzz

code intelligence

Step 1: Find a relevant open source project

Step 2: Understand if the project has potential for it to be fuzzed

Step 3: Make the target compilable with fuzzer instrumentation

Step 4: implement a fuzzer for the given project

Step 5: Integrate the project into the OSS-Fuzz infrastructure.

Step 6: merge your code into OSS-Fuzz Github repository.

Next steps: collect integration reward and continue work

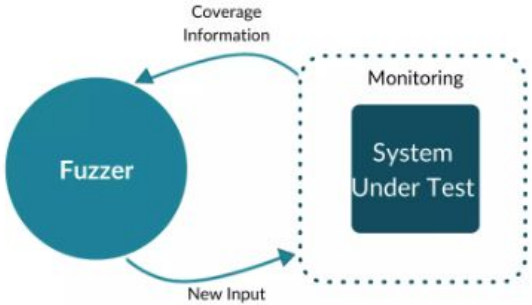
https://adalogics.com/blog/fuzzing-100-open-source-projects-with-oss-fuzz

How does it work?

code intelligence

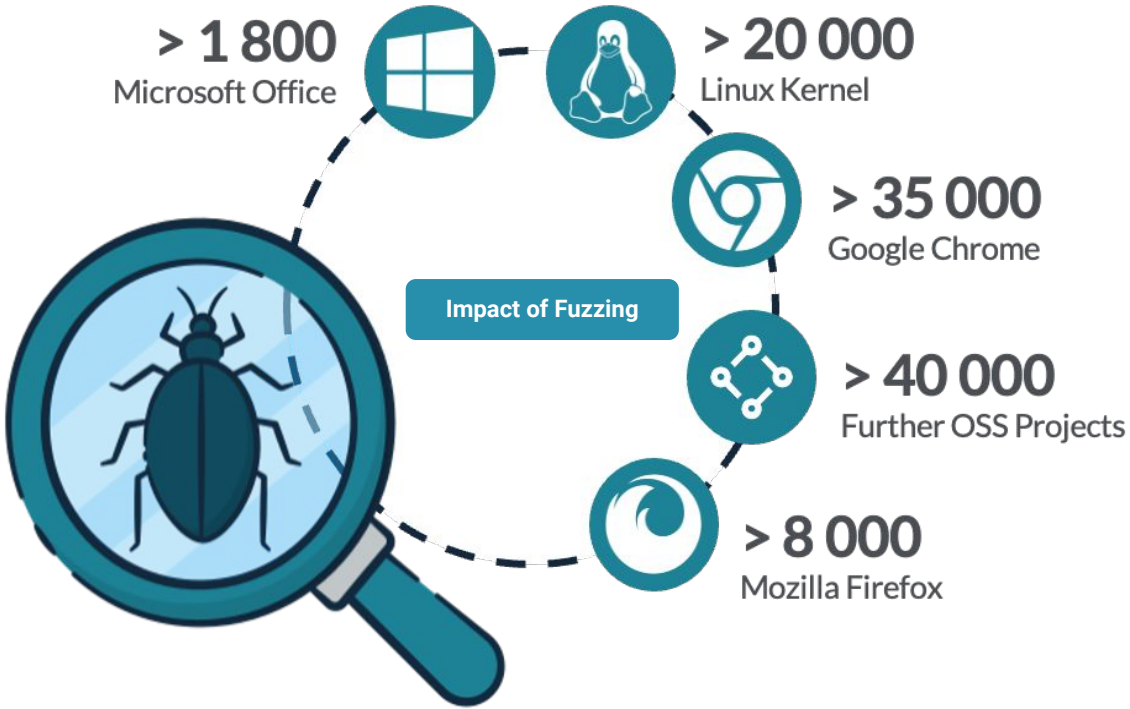
The secret ingredient: Fuzz Testing

Fuzz Testing is a highly scalable application security testing method which helps developers to find **functional bugs** and **security issues** in software.



Fuzz Testing most promising approach for tech leaders and OSS

code intelligence



What bugs can you find with fuzzing? (1/2)

code intelligence

CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer
CWE-823	Use of Out-of-range Pointer Offset
CWE-786	Access of Memory Location Before Start of Buffer
CWE-680	Integer Overflow to Buffer Overflow
CWE-466	Return of Pointer Value Outside of Expected Range
CWE-787	Out-of-bounds Write
CWE-125	Out-of-bounds Read
CWE-129	Improper Validation of Array Index
CWE-131	Incorrect Calculation of Buffer Size
CWE-193	Off-by-one Error
CWE-195	Signed to Unsigned Conversion Error
CWE-839	Numeric Range Comparison Without Minimum Check
CWE-843	Access of Resource Using Incompatible Type ('Type Confusion')
CWE-1257	Improper Access Control Applied to Mirrored or Aliased Memory Regions
CWE-1260	Improper Handling of Overlap Between Protected Memory Ranges
CWE-190	Integer Overflow or Wraparound
CWE-20	Improper Input Validation
CWE-415	Double Free
CWE-416	Use After Free
CWE-476	NULL Pointer Dereference
CWE-590	Free of Memory not on the Heap
CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization
CWE-364	Signal Handler Race Condition
CWE-366	Race Condition within a Thread
CWE-367	Time-of-check Time-of-use (TOCTOU) Race Condition
CWE-368	Context Switching Race Condition
CWE-421	Race Condition During Access to Alternate Channel
CWE-1223	Context Switching Race Condition
CWE-662	Improper Synchronization
CWE-758	Reliance on Undefined, Unspecified, or Implementation-Defined Behavior
CWE-562	Return of Stack Variable Address
CWE-587	Assignment of a Fixed Address to a Pointer
CWE-588	Attempt to Access Child of a Non-structure Pointer
CWE-1102	Reliance on Machine-Dependent Data Representation
CWE-1103	Use of Platform-Dependent Third Party Components
CWE-1105	Insufficient Encapsulation of Machine-Dependent Functionality

https://www.code-intelligence.com/blog/what-bugs-can-you-find-with-fuzzing

What bugs can you find with fuzzing? (2/2)

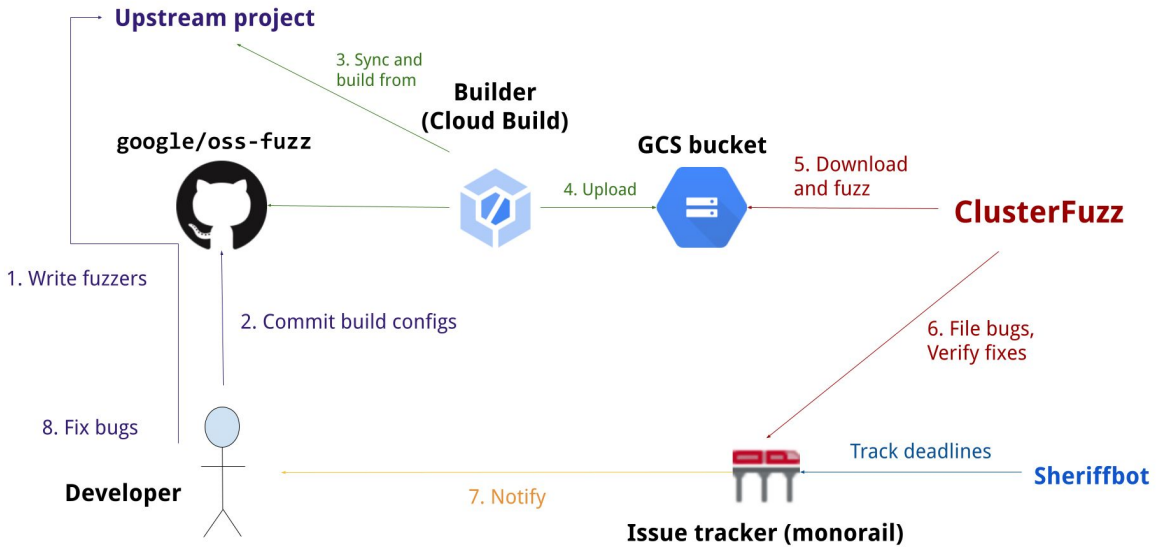
A1: 2017	Injection
A2: 2017	Broken Authentication
A3: 2017	Sensitive Data Exposure
A4: 2017	XML External Entities (XXE)
A5: 2017	Broken Access Control
A6: 2017	Security Misconfiguration
A7: 2017	Cross-Site Scripting XSS
A8: 2017	Insecure Deserialization
A9: 2017	Using Components with Known Vulnerabilities
A10: 2017	Insufficient Logging & Monitoring

CWE-79	Improper Neutralization of Input During Web Page Generation
CWE-1275	Sensitive Cookie with Improper SameSite Attribute
CWE-1004	Sensitive Cookie Without 'HttpOnly' Flag
CWE-614	Sensitive Cookie in HTTPS Session Without 'Secure' Attribute
CWE-778	Insufficient Logging
CWE-779	Logging of Excessive Data
CWE-200	Exposure of Sensitive Information to an Unauthorized Actor
CWE-209	Generation of Error Message Containing Sensitive Information

https://www.code-intelligence.com/blog/what-bugs-can-you-find-with-fuzzing

Google provides a free fuzzing infrastructure for open-source projects

code intelligence

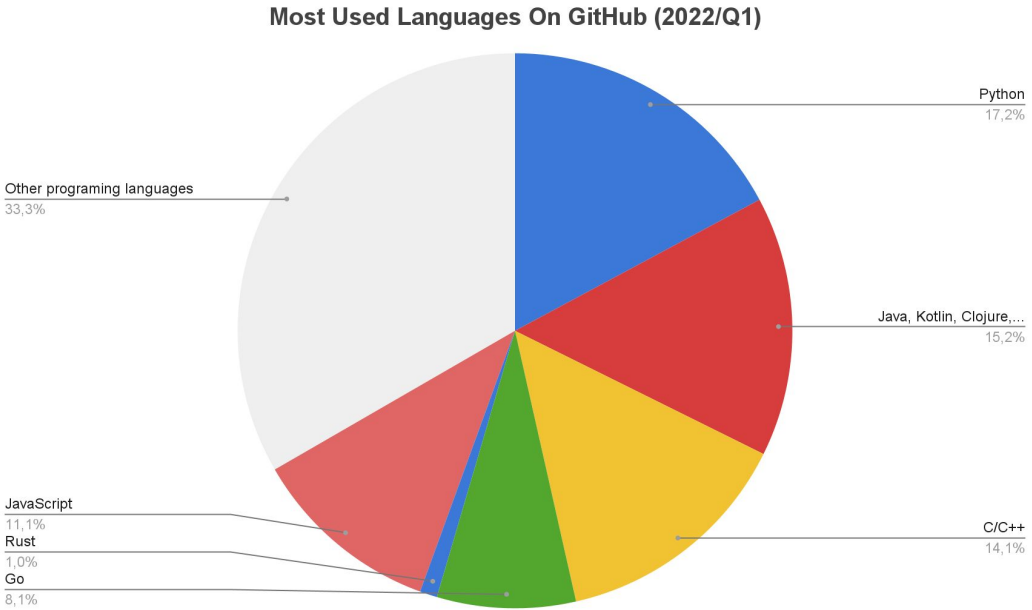


https://github.com/google/oss-fuzz

The service supports common programming languages

code intelligence

We will add language support for even more programming languages to OSS-Fuzz soon

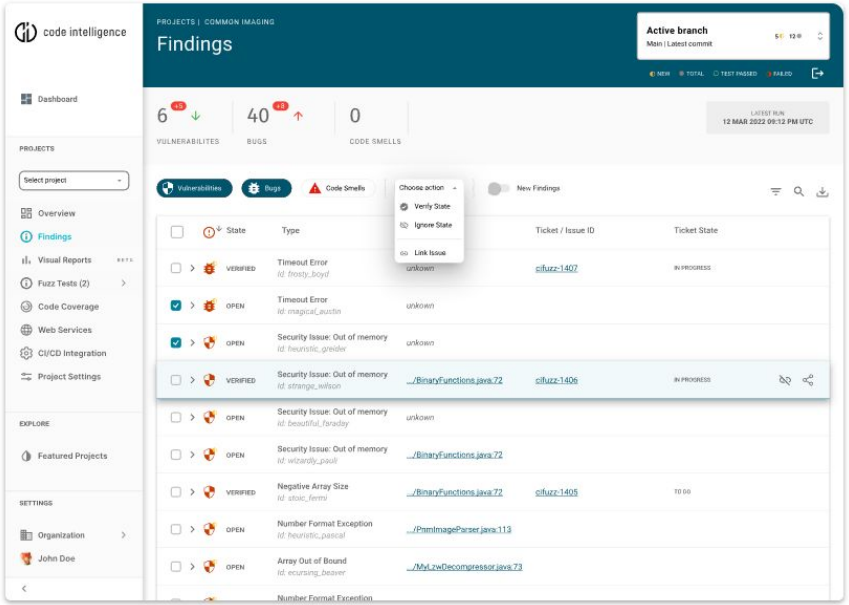


https://madnight.github.io/githut/#/pull_requests/2022/1

How about closed-sourced projects?

code intelligence

Code Intelligence offers a security testing platform with fuzzing technologies, optimized for enterprise use-cases and collaboration in large teams



- Works in your development environment
- Enables you to set up continuous fuzz tests in minutes
- As SaaS or On Premise

https://www.code-intelligence.com/

Contact Details

Jacob Loring

Code Intelligence

info@code-intelligence.com

www.code-intelligence.com

Trusted by