LibreOffice: Software Quality and Security

LibreOffice started life as a fork of OpenOffice.org, the free office productivity suite developed by Sun Microsystems. OpenOffice.org project was born in 2000, one year after the acquisition of StarDivision (the German home of StarOffice) by Sun Microsystems, when Sun itself decided to transform the proprietary suite into an open source suite under the GPL and SISSL licences.

In 2010, the leaders of the volunteer community of the OpenOffice.org project, worried by Sun's management – based on outdated development methodologies and an excessively manual Quality Assurance process – and by Oracle's acquisition of Sun Microsystems (Oracle never hid its idiosyncrasy for open source software) decided to launch an independent project: LibreOffice.

Nevertheless, the quality and security of OpenOffice.org back then was already superior to that of any proprietary software, and in particular Microsoft Office. The CVE (Common Vulnerabilities and Exposures) database reports an order of magnitude higher number of problems, due to two factors: the greater fragility of proprietary source code, which does not benefit from the virtuous effects of security knowledge sharing, and the greater number of users, which makes it an easier target.

The higher quality of open source software was confirmed by Coverity Scan's Open Source Report: “In 2013, the quality of open source projects surpassed that of proprietary projects at all levels. For the 2013 report, we analysed about 500 million lines of code from about 500 proprietary projects written in C/C++ and found that open source software has a lower defect density than proprietary software. One of the factors that led to this result is the effort made by some large projects - including LibreOffice - to collectively resolve more than 11,000 defects during the year” [1]

The Quality of LibreOffice Source Code

When the LibreOffice project was born, the developers changed the approach from OpenOffice.org, launching a source code clean-up activity that lasted throughout 2011, and since the beginning of 2012 has resulted in a significantly better quality office suite. As part of the clean-up activity, the developers also revised their approach to quality assurance, setting up an automated process based on state-of-the-art technologies.

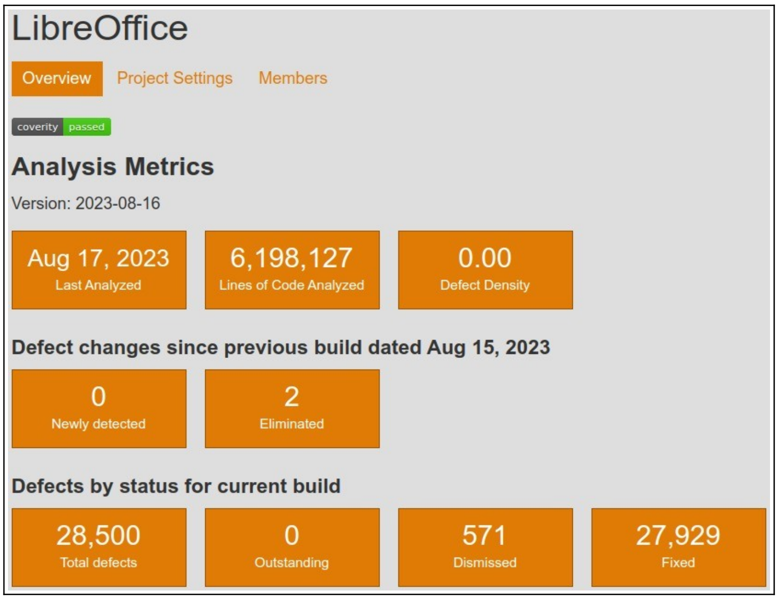
The LibreOffice project uses Gerrit as a patch review tool for its integration with Git, the main distributed system for managing software development. The source code is regularly compiled by a battery of several Tinderboxes, and if the compilation is successful, it undergoes a series of automated tests that verify the behaviour of the software with thousands of documents.

Test files are scraped from several public Bugzilla instances: The Document Foundation, Launchpad (some), Freedesktop, Mozilla, GNOME, KDE, Gentoo, Mandriva, Novell, AbiSource, and W3C SVG test archives. The most suitable documents for testing are the bad ones, so we load and save all of those attached to bugs.

This automated activity is complemented by the work of LibreOffice’s Quality Assurance team, which uses tools such as Bugzilla to manage both bugs and regressions, and to report them to the developers where appropriate for fixing the source code.

Managing Defects in LibreOffice Source Code

The quality of source code has improved significantly since developers started using Coverity Scan services back in 2012 [2]. Since then, LibreOffice has come to be one of the software



LibreOffice

Overview | Project Settings | Members

coverity passed

Analysis Metrics

Version: 2023-08-16

Aug 17, 2023	6,198,127	0.00
Last Analyzed	Lines of Code Analyzed	Defect Density

Defect changes since previous build dated Aug 15, 2023

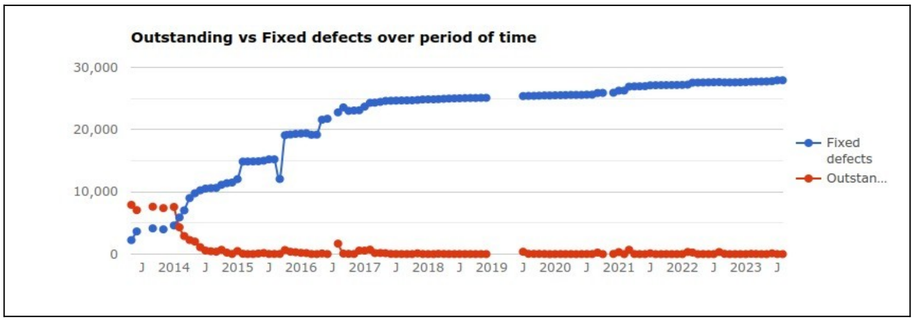
0	2
Newly detected	Eliminated

Defects by status for current build

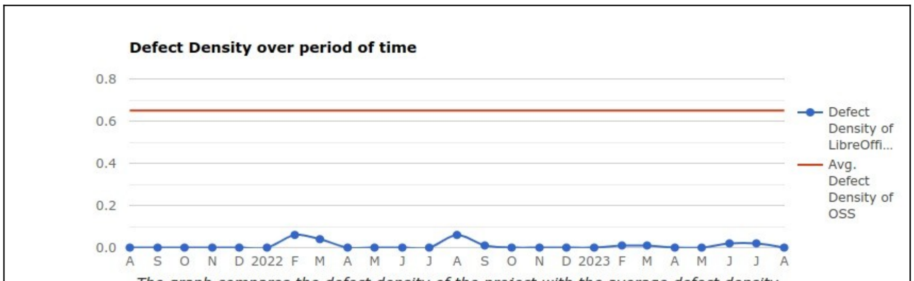
28,500	0	571	27,929
Total defects	Outstanding	Dismissed	Fixed

packages with the fewest defects as a proportion of source code lines. This activity is very important in terms of software security, as defects in source code are often associated with CVE reports (common vulnerabilities and exposures).

The first image provided by Coverity Scan represents the current situation of LibreOffice 7.6 Community's source code, with 0 outstanding defects. Over time, LibreOffice developers have fixed 27,929 defects, while 571 defects were dismissed as false positives.



The second image provided by Coverity Scan summarises the trend of outstanding versus fixed defects during the last 10 years. Since 2015, the number of outstanding defects has steadily been 0 or close to 0, while the number of fixed defects has been regularly increasing (the gap in 2019 is due to a complete overhaul of Coverity Scan analysis software).



The graph compares the defect density of the project with the average defect density of open source projects that are similar in size (i.e. more than 1 million lines of code)

This third image provided by Coverity Scan provides a better visualization of the defect density trend over the last couple of years, from August 2021 to August 2023. Only in February/March and August 2022 was LibreOffice defect density above 0.005 defects per 1,000 lines of code.

The numbers provided by Coverity Scan are a testament to the cleaning and refactoring of LibreOffice's source code by developers since 2010. They also confirm the extent of the technical debt inherited from OpenOffice.org, which was completely resolved within four years. It was valuable work, understood by the market only in retrospect, when it finally became clear that LibreOffice's development strategy is appropriate.

DEFECTS WE FIND INCLUDE
Crashes
Security Vulnerabilities
Concurrency
Memory Corruption
Uninitialized memory
Error handling
Resource leaks

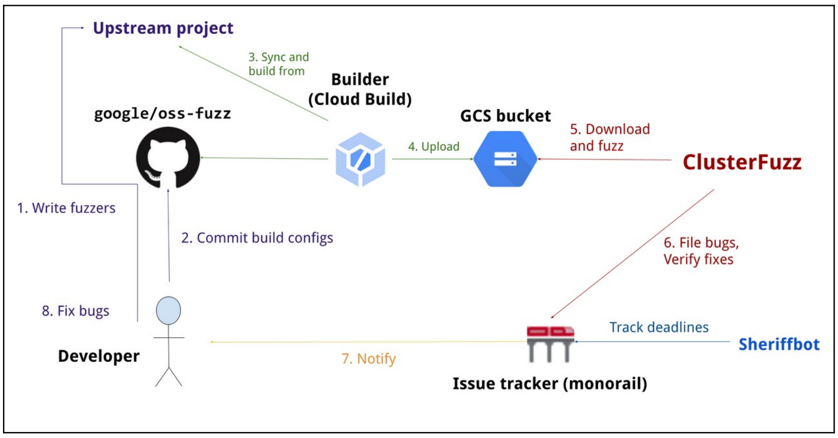
Coverity Scan allows for the detection of crashes, security vulnerabilities, concurrencies, memory corruption, uninitialized memory, error handling, and resource leaks, and contributes to overall software security, because reducing the number of defects in the source code provides developers a more robust and resilient foundation for their work.

It is worth emphasizing, however, that a low number of defects on the source code - such as the one in LibreOffice - does not necessarily preclude the presence of bugs, regressions, and vulnerabilities.

Fuzzing to test LibreOffice source code

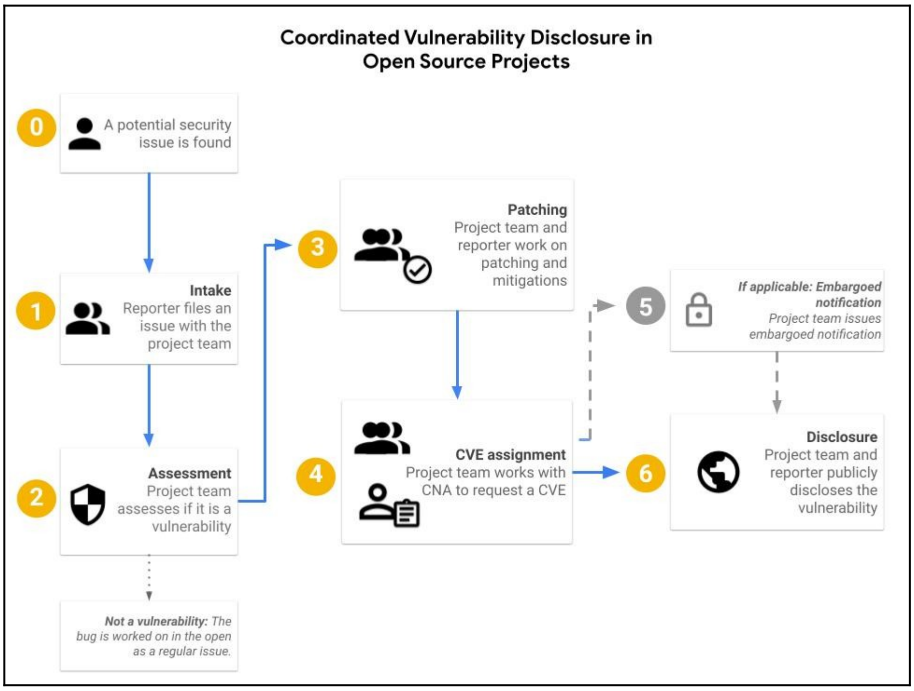
Fuzzing is an automated software testing technique, extensively used by LibreOffice developers, that provides invalid, unexpected, or random data as inputs to a computer program. The application is then monitored by the code sanitizer, a programming tool that detects bugs in the form of undefined or suspicious behavior, for issues such as crashes, failing built-in code assertions, or potential memory leaks.

Fuzzing is used as an automated technique to expose the vulnerabilities in security-critical programs that might be exploited with malicious intent, to demonstrate the presence of bugs rather than their absence.



The main fuzzing tool adopted by LibreOffice developers is Google OSS Fuzz, announced in 2016. It is a testing infrastructure used for Chrome and other free and open source software (FLOSS) projects, which combines fuzzing engines with sanitizers and provides a massive distributed execution environment powered by ClusterFuzz. Using OSS-Fuzz, the project is fuzzing 50 file formats and that fuzzing is constantly running as new changes are merged.

Managing Vulnerabilities in LibreOffice’s Source Code


Coordinated Vulnerability Disclosure in Open Source Projects
0
A potential security issue is found
1
Intake
Reporter files an issue with the project team
2
Assessment
Project team assesses if it is a vulnerability
Not a vulnerability: The bug is worked on in the open as a regular issue.
3
Patching
Project team and reporter work on patching and mitigations
4
CVE assignment
Project team works with CNA to request a CVE
5
If applicable: Embargoed notification
Project team issues embargoed notification
6
Disclosure
Project team and reporter publicly discloses the vulnerability

The Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information around security vulnerabilities and exposures. The United States' National Cybersecurity FFRDC, operated by The MITRE Corporation, maintains the system, with funding from the US National Cyber Security Division of the US Department of Homeland Security.

A vulnerability is a computer software system's weakness that enables unwarranted access. The CVE Identifier is the unique number assigned to each vulnerability by a CVE Numbering Authority (CNA), such as The Document Foundation in the case of LibreOffice. When investigating a vulnerability or potential vulnerability it helps to acquire a CVE number early on, as all future correspondence can refer to it.

According to MITRE Corporation CVE database, which can be found at https://www.cve.org/ (formerly at https://cve.mitre.org/), LibreOffice has been affected by 50 CVEs during the last 10 years. In the same period, Microsoft Office was affected by 505 CVEs - so one order of magnitude higher than LibreOffice.

In addition, all CVEs affecting LibreOffice were resolved with a patch released prior to the disclosure (by convention, the publication of CVEs in the database occurs between 30 and 60 days after the problem is reported to the security team of the applications affected).

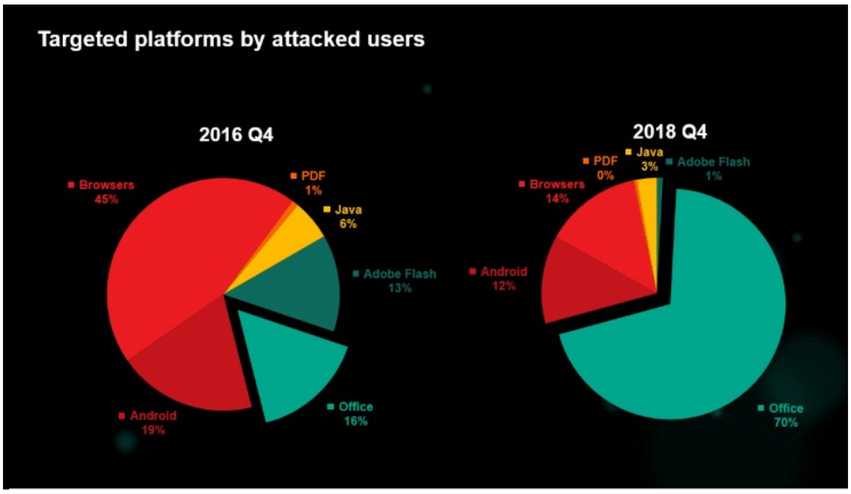
The website https://www.cvedetails.com/ provides a comparison based on the Common Vulnerability Scoring System (CVSS), an open set of standards used to assess a vulnerability and assign a severity along a scale of 0-10 (none to critical).

LibreOffice’s ESC (Engineering Steering Committee) is supported by a team of experts in more specific security issues, with world-class specialists, who often volunteer as security experts for companies that develop software in other sectors (a typical example is automotive software).

Having said all this, it should be made clear that the number of vulnerabilities can't be used to define a program's competitive advantage, and in fact we have never mentioned it to date, because it does not define the quality of the application itself but only represents the theoretical risk of it being used to access user data or compromise PC security. Thus, users should never choose one software over another based on the number of vulnerabilities.

The importance of LibreOffice's native ODF file format

LibreOffice uses the open standard Open Document Format (ODF) as its native format, which can help organisations and companies to reduce their vulnerability to attacks from outside, compared to what can happen with proprietary file formats.



Proprietary office document formats are one of the most exploited vulnerabilities, according to independent research conducted by Symantec in 2011 and Kaspersky Labs in 2018. At the 2019 Security Analyst Summit, Kaspersky said that around 70 percent of all attacks detected in Q4 2018 were trying to abuse a Microsoft Office vulnerability, a dramatic increase versus the 16 percent detected in 2016 (the slide is from the original presentation) [3].

The explanation is simple. Proprietary file formats such as the legacy DOC, XLS and PPT, and the current “transitional” DOCX, XLSX and PPTX, can contain binary blobs of data – which are the preferred vehicle for malware – to allow backward compatibility with old documents, a feature which was meant to protect users from content obsolescence.

Backwards compatibility, and its related binary blobs, not only reduce the security of Microsoft Office documents, but also prevent them from being standards compliant. In fact, while Office Open XML “strict” specifications do not foresee the integration of binary blobs as they cannot be visually represented by the XML code, they are allowed by the current “transitional” file format, which further adds complication and risk.

In contrast, the introduction of ODF created a break in the backward compatibility of documents, which was solved with software tools for format conversion. In this way, the format has always consistently adhered to the description of the XML-based standard, and has never required the integration binary blobs.

Of course, the use of the standard ODF format cannot guarantee the security of software, although it can simplify the task of the tools that must check for malicious code. The protection of users and their stakeholders is left to the security measures and anti-virus programs adopted by the individual or the organization.

In the case of LibreOffice, the standard Open Document Format is an important element which complements the work of the team of security experts by reducing the attack surface. LibreOffice’s security is the result of a global effort of the entire community, from companies in the ecosystem to volunteers contributing to development, Quality Assurance, documentation and localization.

Credits

LibreOffice’s security is the result of a huge amount of work by a group of people led by Caolán McNamara, which collectively adds up to some significant and ongoing support of LibreOffice by several brand-names:

- Red Hat, which led security activities for years
- Collabora, which has inherited Red Hat's leadership on security
- allotropia, which is supporting security-related efforts in specific areas
- Google, which is sponsoring OSS Fuzz and providing a lot of expensive CPUs
- Synopsys, which is providing the Coverity Scan static code analyzer for free
- Adfinis, which is funding the hardware for the crash tests
- The Document Foundation, which is providing the infrastructure for development and especially Quality Assurance, including professionals to coordinate activities

In addition, LibreOffice’s security is also related to the incredible work of volunteers in development and Quality Assurance, and to many contributions from security-focused companies such as Forcepoint.

Notes

[1] https://www.zdnet.com/article/coverity-finds-open-source-software-quality-better-than-proprietary-code/

[2] https://scan.coverity.com/projects/libreoffice. Current numbers are based on the analysis of LibreOffice 24.2 source code, which will be released in early February 2024. At that time we will update the document based on the new figures.

[3] https://www.zdnet.com/article/kaspersky-70-percent-of-attacks-now-target-office-vulnerabilities/