MEND

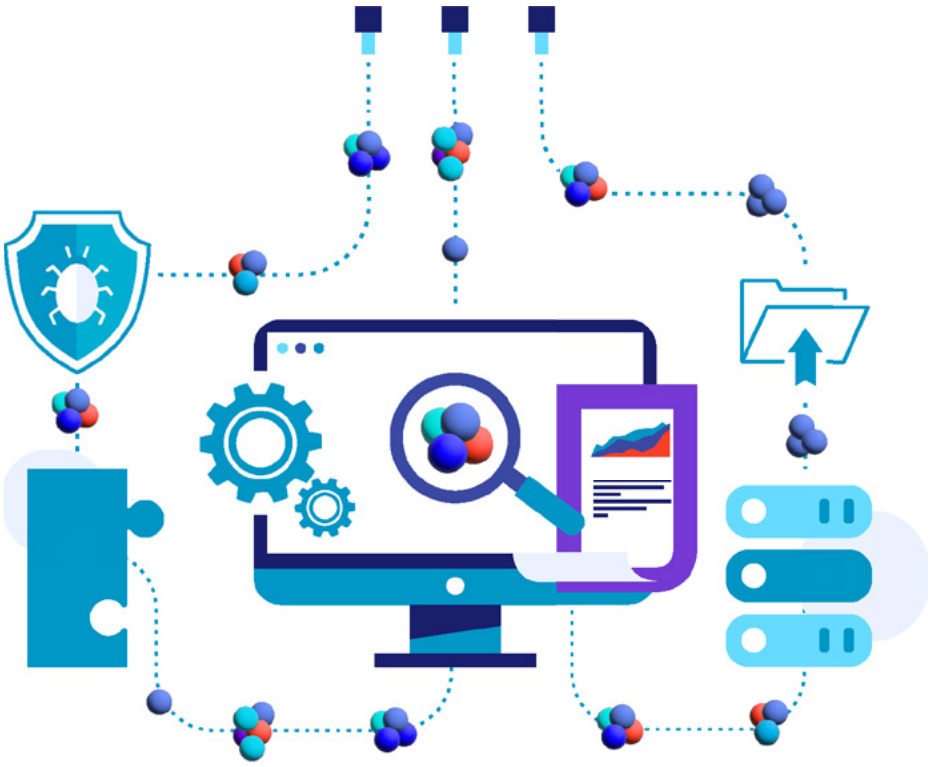
The State of Open Source Security Vulnerabilities

Annual Report 2021

2020 presented us all with a set of challenges no one could have expected. The pandemic initially raised a lot of uncertainty in the software development industry. Companies pivoted to remote work practically overnight and faced a series of issues encompassing everything from application security to employee well-being.

Shifting to work from home introduced new security threats. Early on many organizations' budgets were put under scrutiny, and many worried that investment plans for application security strategies would be put on the back burner while many industries went into survival mode.

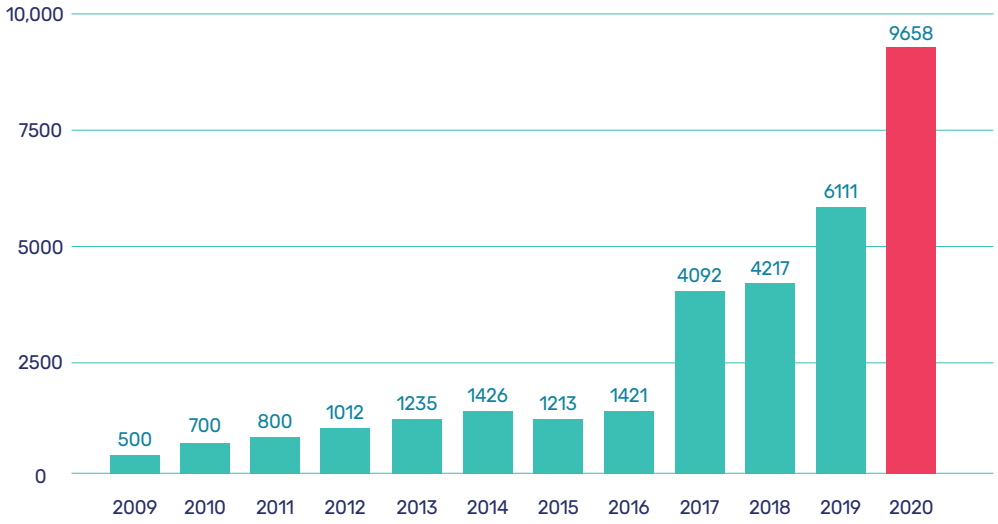
In this report, we analyzed Mend's open source vulnerabilities database to gain insights on the state of open source security and learn how to best address the challenge of developing secure software products at the speed of DevOps.



The Number of Open Source Vulnerabilities Continues to Rise

According to the Mend database, aggregated from the NVD, dozens of security advisories, peer-reviewed vulnerability databases, and popular open source issue trackers, the number of published open source software vulnerabilities in 2020 rose once again, by over 50%.

Open Source Vulnerabilities per Year: 2009-2020



The Open Source Development and Security Communities: More Active Than Ever

There are a few possible explanations to the sharp increase in the number of known open source vulnerabilities in 2020.

First is **increased activity in the open source community**. While no one was sure how shifting to remote work would effect developers, it appears that in the first months of the pandemic open source developers were working harder than ever. GitHub reported a sharp increase in open source project creation in March and April 2020. This rise in activity most probably extended to more open source security research.

The addition of CVE Numbering Authorities (CNAs) also contributed to the increase in published vulnerabilities. GitHub Security Labs, launched over a year ago, invested a lot of effort in detecting, fixing, and publishing vulnerabilities in open source components. Since then more software development organizations have joined the open source community's efforts to ensure issues are analyzed, fixed, and published as soon as possible.

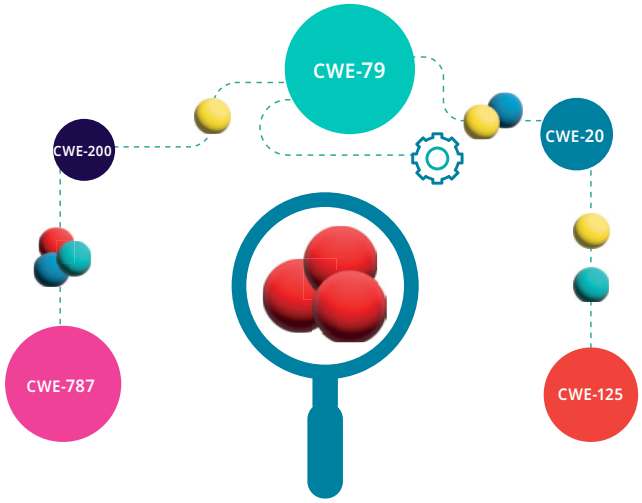
In addition to the many hands on deck, **automation** also explains the high number of open source vulnerabilities discovered this year. Security researchers are using automated scanning and detection tools to find vulnerabilities, enabling them to find and fix security issues quickly, and at a greater volume.

In some cases, researchers found multiple CVEs where a common problem affected many projects. Sometimes a single CVE was applied to many projects but other times it was correct for them to each have their own CVE.

Most Common CWEs in Open Source Components

As AppSec continues to shift left into the design and development phases and responsibility over security is shared with developers, secure coding practices and tools become an important part of the DevSecOps pipeline.

In order to gain insights on secure coding with open source components, we decided to dive deep into the data on the most common CWEs in vulnerable open source components detected in 2020.



Open Source Vulnerabilities in 2020: Top CWEs

2015	2016	2017	2018	2019	2020
CWE-79 XSS	CWE-119 Buffer Overflow	CWE-119 Buffer Overflow	CWE-79 XSS	CWE-79 XSS	CWE-79 XSS
CWE-119 Buffer Overflow	CWE-264 Permissions, Privileges, and Access Controls	CWE-79 XSS	CWE-190 Integer Overflow	CWE-20 Improper Input Validation	CWE-787 Out-of-bounds Write
CWE-264 Permissions, Privileges, and Access Controls	CWE-20 Improper Input ValiAdation	CWE-125 Out-of-bounds Read	CWE-119 Buffer Overflow	CWE-125 Out-of-bounds Read	CWE-125 Out-of-bounds Read
CWE-200 Information Exposure	CWE-200 Information Exposure	CWE-200 Information Exposure	CWE-20 Improper Input ValiAdation	CWE-89 SQL Injection	CWE-20 Improper Input ValiAdation
CWE-20 Improper Input ValiAdation	CWE-79 XSS	CWE-20 Improper Input ValiAdation	CWE-125 Out-of-bounds Read	CWE-325 CSRF	CWE-200 Information Exposure

Open Source Vulnerabilities in 2020: Top CWEs

While CWE-79 (Cross-site scripting) has been at the top of the list for the past few years, CWE-787 is a new arrival to the top five. It might seem like CWE-787 came out of nowhere, but it's actually a descendent of the common CWE-119 (Buffer overflow), which saw a decrease this year. We can also see that CWE-125, another child of CWE-119, is also a prominent issue.

It appears there was an effort to map CVEs directly to weaknesses like CWE-787 and CWE-125 instead of categories like CWE-119. This included a large remapping effort of over 10,000 CVE entries. Improper Input Validation and Information Exposure are other examples of categories that are being remapped into the more precise weaknesses.

1	CWE-79	XSS
2	CWE-787	Out-of-bounds Write
3	CWE-125	Out-of-bounds Read
4	CWE-20	Improper Input Validation
5	CWE-200	Information Exposure
6	CWE-416	Use After Free
7	CWE-89	SQL Injection
8	CWE-22	Path Traversal
9	CWE-352	CSRF
10	CWE-190	Integer Overflow

Open Source Vulnerabilities in 2020: Top CWEs

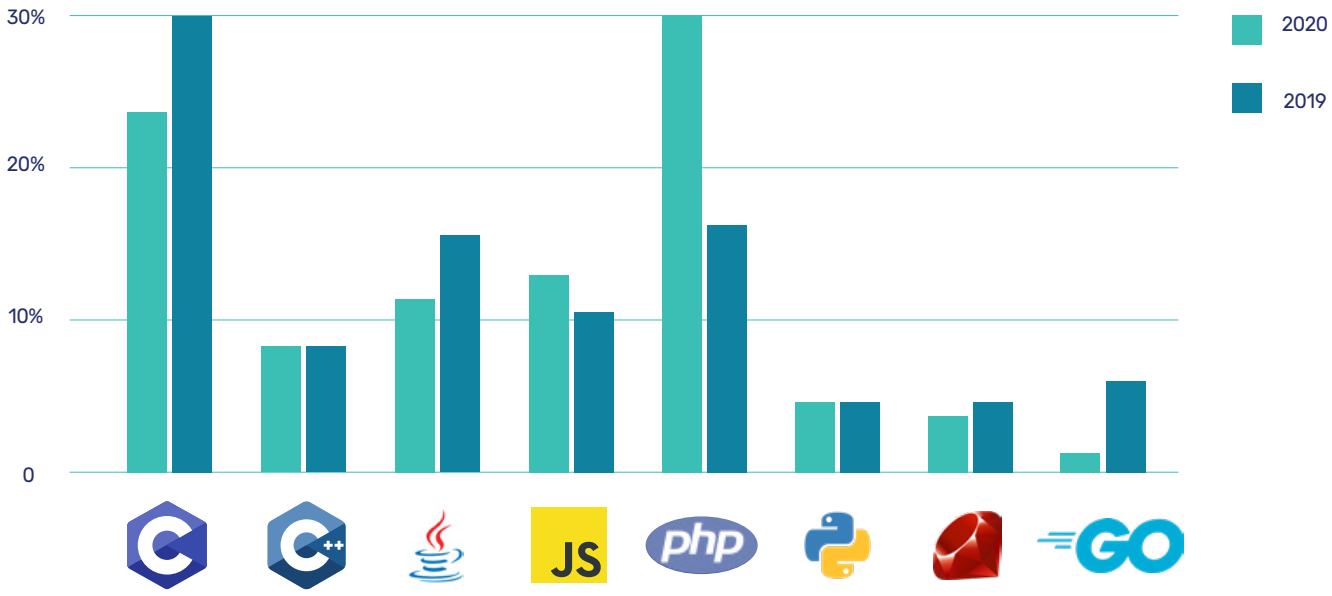
Continuing our research into secure coding, we also looked at some of the top programming languages, including how many and what type of open source security vulnerabilities were disclosed per language.

Top CWEs per Programming Language 2020:

	CWE-79 XSS	CWE-862 Missing Authorization	CWE-502 Deserialization of Untrusted Data
	CWE-79 XSS	CWE-20 Improper Input Validation	CWE-787 Out-of-bounds Write
	CWE-79 XSS	CWE-89 SQL Injection	CWE-352 Cross-Site Request Forgery
	CWE-79 XSS	CWE-20 Improper Input Validation	CWE-200 Information Exposure
	CWE-200 Information Exposure	CWE-79 XSS	CWE-863 Improper Input Validation
	CWE-787 Out-of-bounds Write	CWE-125 Out-of-bounds Write	CWE-476 NULL Pointer Dereference
	CWE-732 Incorrect Permission Assignment for Critical Resource	CWE-200 Information Exposure	CWE-20 Improper Input Validation

We checked which CWEs were most prominent in some of the most popular programming languages. Cross-site scripting (CWE-79) continues to dominate in most of the programming languages that we looked at, especially those used for web development. Another reason for how common XSS issues are is that they are very easy to detect using automated tools. We also see that many of the newly discovered CWE-787 Out-of-bounds write vulnerabilities were discovered in C.

Vulnerabilities in Top Programming Languages: 2020 vs. 2019



The increase in Buffer overflow related issues is one of the reasons that C saw so many new vulnerabilities in 2020.

According to GitHub's 2020 Octoverse, PHP's popularity is decreasing in the open source community. The decrease in the number of vulnerabilities in PHP is probably a result of decreased community interest.

Go, on the other hand, is gaining popularity along with increased security research. Last year Go vulnerabilities amounted to only 1% of vulnerabilities in all programming languages, compared to 5% in 2020.

Go is a relatively young language, and the rising number of vulnerabilities discovered in Go might also be because many of the Go projects are written from scratch, rather that using open source libraries and components that have been under the security microscope for years.

Open Source Vulnerabilities: Severity Breakdown

Addressing the sharp rise in the number of open source vulnerabilities published this year is a major challenge for software development organizations. As security debt continues to rise for most, it's important to find a way to prioritize vulnerabilities remediation. One parameter that many organizations look to when attempting to decide what to remediate first is the vulnerabilities' severity score.

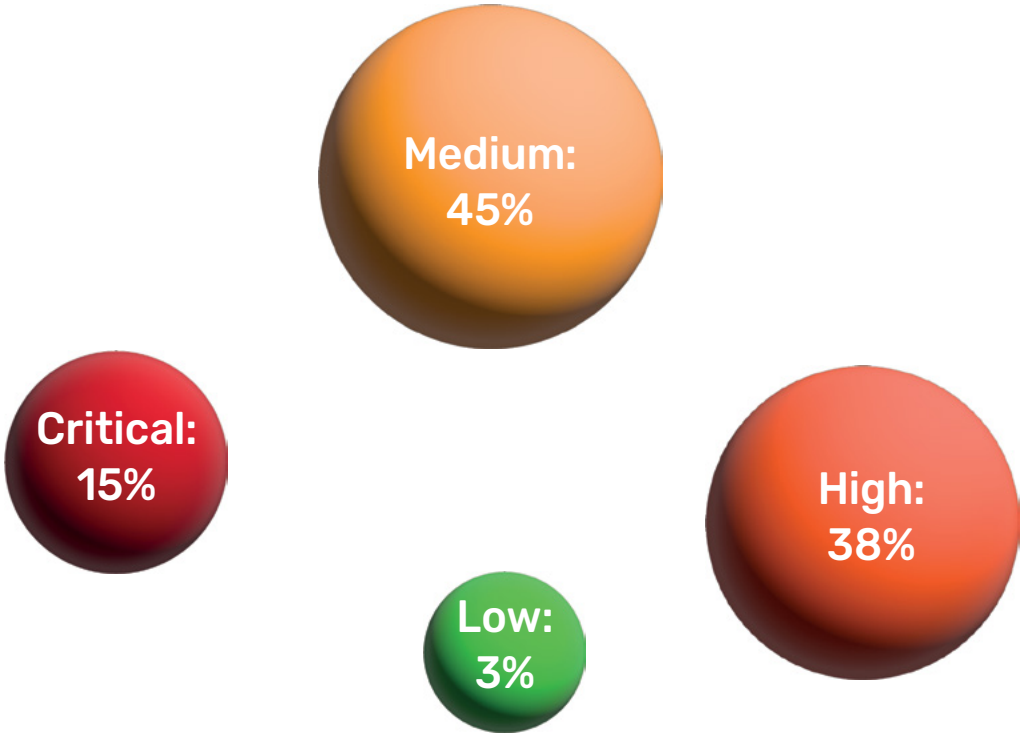
We checked the breakdown of open source vulnerabilities' severity scores to see if this is an effective technique.

Open Source Vulnerabilities in 2020: Severity Breakdown

The fact that over 50% of new open source security vulnerabilities are rated high or critical doesn't help security and development teams that rely on severity scores when considering which issues to address first.

Fixing all issues, or even "only" high and critical issues, is an unrealistic plan for teams that want to keep up with the rapid pace of development.

Organizations need to leverage prioritization and remedian tools that target the vulnerabilities that will most impact their systems and business if they want to manage their security debt wisely.



Final Thoughts

Despite how tumultuous 2020 turned out to be, it turns out application security -- and open source security in particular -- remains a top concern and priority for both the software development industry and the open source community.

The sharp increase in the overall number of open source vulnerabilities published in 2020 is another reminder that open source security must be addressed as an integral part of an organization's AppSec strategy, and requires organizations adopt a set of security practices. These include auto-remediation and a vulnerabilities prioritization strategy, ensuring that the issues that pose the biggest threat are addressed first.

About Mend

Mend, formerly known as WhiteSource, effortlessly secures what developers create. Mend uniquely removes the burden of application security, allowing development teams to deliver quality, secure code, faster. With a proven track record of successfully meeting complex and large-scale application security needs, the world's most demanding software developers rely on Mend. The company has more than 1,000 customers, including 25 percent of the Fortune 100, and manages Renovate, the open- source automated dependency update project.

For more information, visit www.mend.io, the Mend blog, and Mend on LinkedIn and Twitter.