WhiteSource

The State of Open Source Security
Vulnerabilities

WhiteSource Annual Report 2020

Open source components have become an integral part of today's software applications — it's impossible to keep up with the hectic pace of release cycles without them. As open source usage continues to grow, so does the number of eyes focused on open source security research, resulting in a record-breaking number of published open source security vulnerabilities in 2019.

This research report focuses on open source security's weakest and strongest points in the hopes of bringing some clarity the fast-paced and complex space of known open source security vulnerabilities.

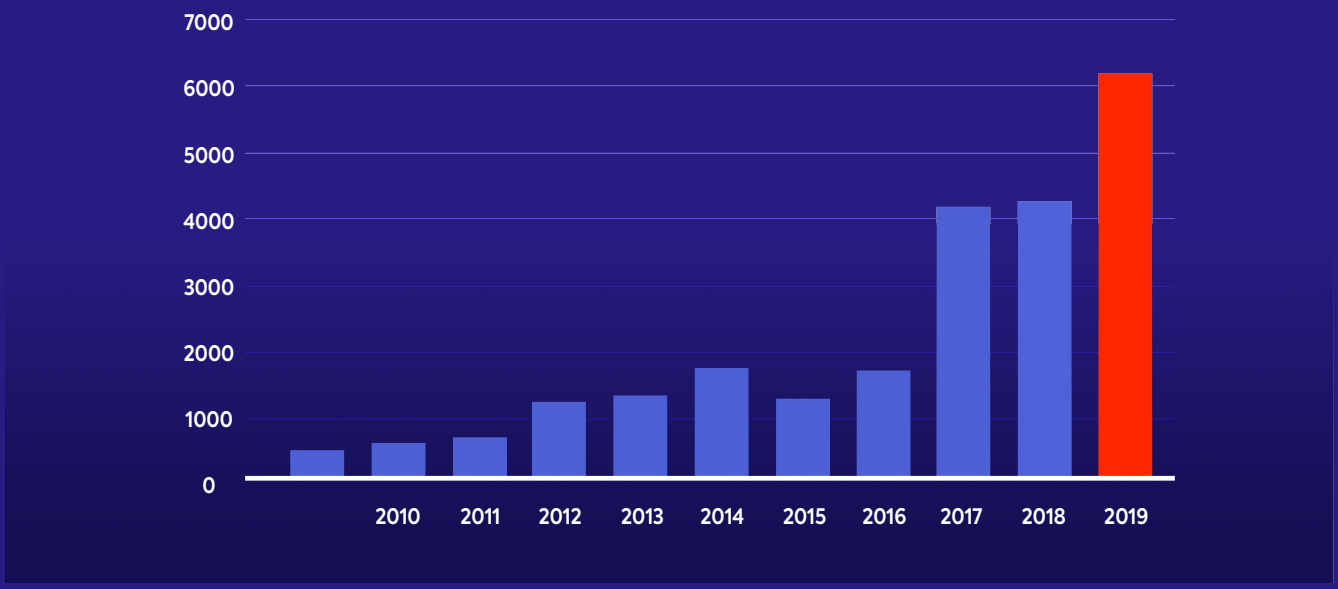
In this report we'll be looking at the number of open source vulnerabilities published this year, and then take a closer look at open source security vulnerabilities in popular programming languages, most common CWEs over the years, open source vulnerabilities' scoring and severity, and vulnerabilities in the most popular open source projects that we all use and love.

Open Source Security Vulnerabilities Are ON THE RISE

According to the WhiteSource database, aggregated from the NVD, dozens of security advisories, peer-reviewed vulnerability databases, and popular open source issue trackers, the number of disclosed open source software vulnerabilities in 2019 skyrocketed to over 6000 reported vulnerabilities.

This can be attributed to the rise in awareness open source security following the widespread adoption of open source components and the massive growth of the open source community over the past few years, along with the media attention directed at recent data breaches. The Number of Reported Open Source Vulnerabilities Rose by Nearly 50% in 2019.

Open Source Security Vulnerabilities per Year



Good News

Over 85% of open source security vulnerabilities are disclosed with a fix already available.

Tech Giants have invested heavily in better securing and managing open source projects over the past few years, and the community is working hard at security research to publish newly discovered open source security vulnerabilities along with a fix.

The fix will usually be an updated version or a patch for the vulnerable code.

Bad News

Unfortunately, users are not always able to benefit from the community's efforts. Only 84% of known open source vulnerabilities appear in the NVD.

Information about vulnerabilities is not published in one centralized location, rather scattered across hundreds of resources, and sometimes poorly indexed — often making searching for specific data a challenge.

While 45% of reported open source vulnerabilities are not initially reported to the NVD, many end up being published in the NVD, months after being reported in other resources. Based on WhiteSource's database, only 29% of all open source vulnerabilities reported outside of the NVD are eventually published in it.

Only 84% of known open source vulnerabilities eventually appear in the NVD.

2020 PREDICTIONS

Given the continued increase of both open source usage and security research, the number of reported open source vulnerabilities will surely keep rising.

In addition, we're starting to see the open source community looking for new initiatives in order to address the chaos in the open source security process. One good example is the GitHub Security Lab, which aims to help researchers, open source project maintainers, and users to easily report suspected vulnerabilities in a secure manner without exposing a zero-day vulnerability into the world.

GitHub's embedded disclosure process will encourage open source project maintainers to properly report vulnerabilities, rather than just push a fix. Having the maintainers themselves report vulnerabilities should also lead to higher-quality metadata, like affected versions and fixed-in versions, as opposed to a third party reporting the problem.

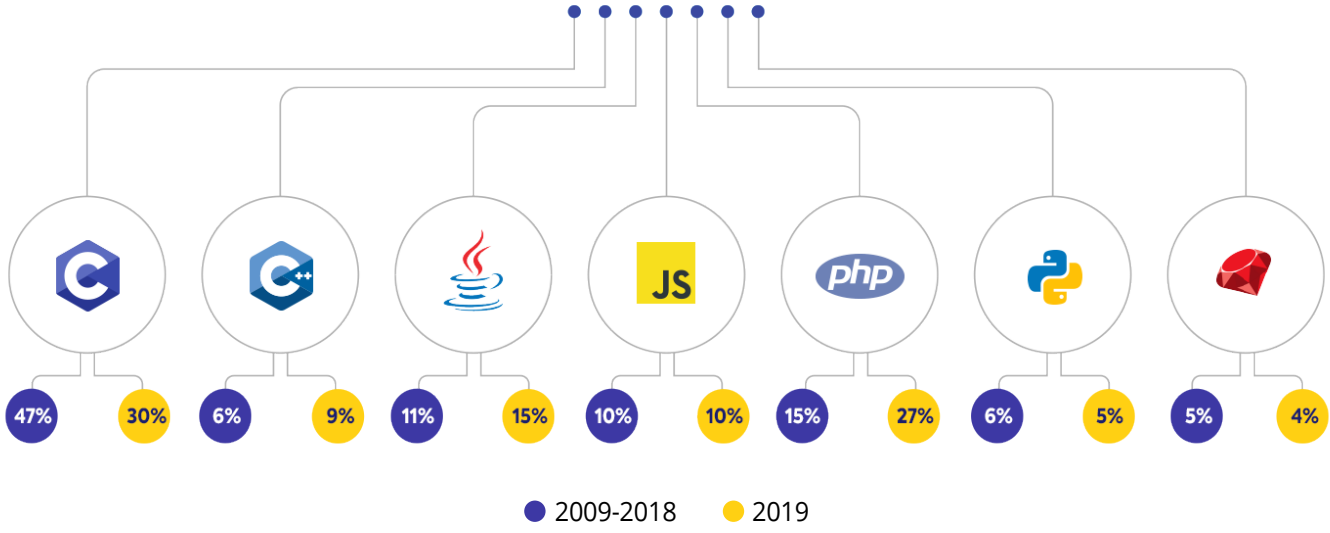
Our concern is that, while these tools will help to report vulnerabilities in a proper manner, they will probably only aggravate the current problem as software developers are already struggling to keep up with the increased rate of reported open source vulnerabilities.

Which Programming Languages Are MOST SECURE?

Looking at the drastic overall rise in the number of vulnerabilities in 2019, we couldn't help but wonder whether it was consistent across the top programming languages.

We compared how the top seven coding languages stack up when it comes to reported open source vulnerabilities in 2019, and then compared those numbers to the past ten years.

Open Source Vulnerabilities per Language, 2019 vs. 2009-2018



C still has the highest percentage of vulnerabilities due to the high volume of code written in this language.

However, the numbers are continuously trending down because other languages are also becoming popular. That said, PHP's relative number of vulnerabilities has risen significantly, while there's no indication of the same rise in popularity.

Shout-out to Python, which still has a relatively low percentage of vulnerabilities, even though its popularity, especially in the open source community, continues to rise. Hopefully, this is a result of secure coding practices and not lax security research for python projects.

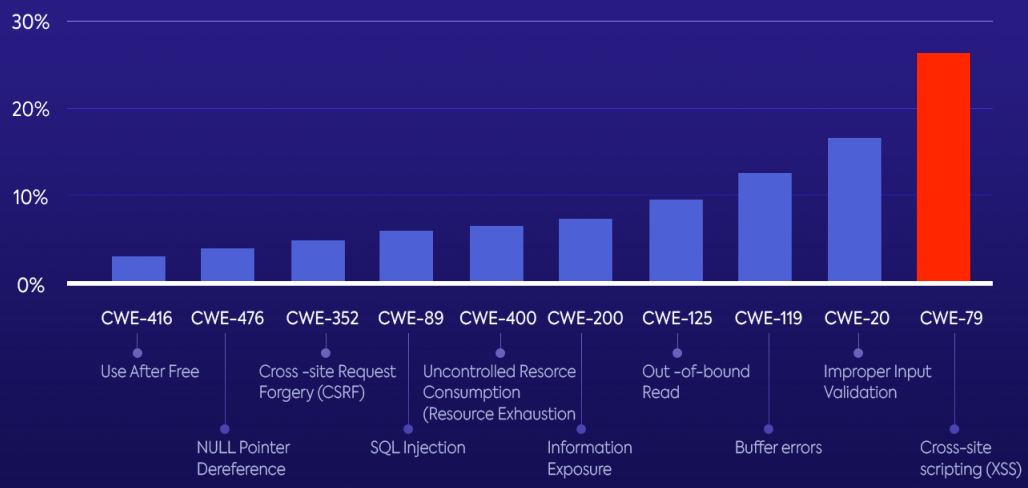
Which CWE's Do We Need To Watch Out For In 2020?

Another aspect we wanted to look at was the types of vulnerabilities that were most common in 2019.

The top five CWE's in 2019 have been consistent over the past several years, and are all related to information disclosure.

What's concerning is that the most common CWE's are due to simple code errors and imprecise coding, that all developers could avoid by sticking to fairly basic coding standards.

Most Common CWE's in 2019



While they are not in the top five, it's interesting that CWE-352 — Cross-Site Request Forgery (CSRF), has emerged in the top 10 CWEs this year, and that CWE-89 — SQL Injection, re-emerged after it wasn't one of the top CWE's since 2015. This might be due to an increase in the volume of open source web projects developed, and it might indicate that web vulnerabilities are on the rise and something we should be mindful of when coding.

Most Common CWE's per Year, 2014-2019

	1	2	3	4	5
2019	CWE-79 Cross-site scripting (XSS)	CWE-20 Improper Input Validation	CWE-119 Buffer Errors	CWE-125 Out-of-bounds Read	CWE-200 Information Exposure
2018	CWE-79 Cross-site scripting (XSS)	CWE-119 Buffer Errors	CWE-20 Improper Input Validation	CWE-125 Out-of-bounds Read	CWE-200 Information Exposure
2017	CWE-119 Buffer Errors	CWE-125 Out-of-bounds Read	CWE-79 Cross-site scripting (XSS)	CWE-200 Information Exposure	CWE-20 Improper Input Validation
2016	CWE-119 Buffer Errors	CWE-20 Improper Input Validation	CWE-200 Information Exposure	CWE-264 Permissions, Privileges, and Access Controls	CWE-284 Improper Access Control
2015	CWE-119 Buffer Errors	CWE-79 Cross-site scripting (XSS)	CWE-264 Permissions, Privileges, and Access Controls	CWE-200 Information Exposure	CWE-20 Improper Input Validation
2014	CWE-79 Cross-site scripting (XSS)	CWE-264 Permissions, Privileges, and Access Controls	CWE-119 Buffer Errors	CWE-20 Improper Input Validation	CWE-399 Resource Management Errors

Most Common CWE's per Year, 2014–2019

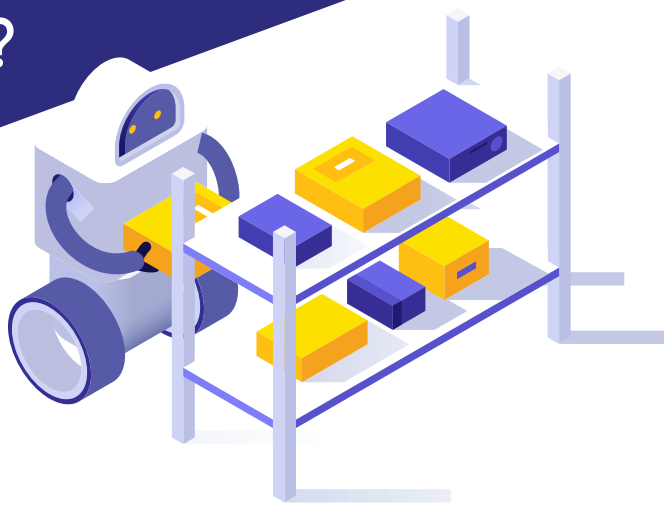
When we examined the top three CWEs for the top programming languages, we noticed that three CWEs were at the top of the list for all of the languages, except C:

- Cross-Site-Scripting (XSS) (CWE-79)
- Information Leak / Disclosure (CWE-200)
- Input Validation (CWE-20)

It is not surprising to see that most of the languages share quite a few of their top CWEs. Researchers are increasingly relying on automated detection tools to find security vulnerabilities, and XSS and Input Validation issues are relatively easy to find with these types of tools.

Another factor that might contribute to the consistency of Information Exposure across most languages is that it's such a general issue that a variety of vulnerabilities are most probably grouped under this category.

VULNERABILITY SEVERITY SCORING: AN OBJECTIVE PRIORITIZATION STANDARD?



The rising number of reported vulnerabilities demands that development teams quickly prioritize their security alerts. The CVSS (Common Vulnerability Scoring System) score is usually the go-to parameter for remediation prioritization, but should it be?

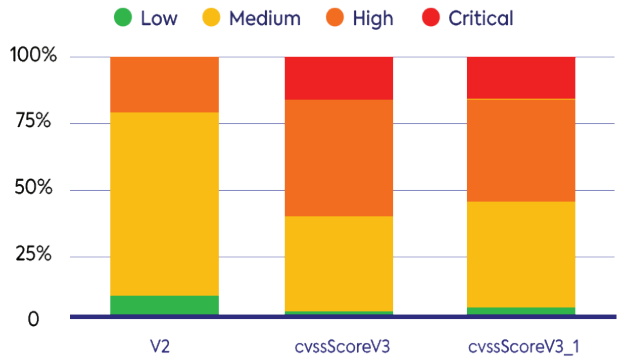
CVSS was updated several times over the past few years (v2 to v3, and most recently v3.1), in the hopes of achieving a measurable, objective standard that helps support all organizations and industries. However, it has also changed the definition of what a high severity vulnerability is.

We looked at over ten thousand vulnerabilities from 2016 to 2019 and checked their CVSS v2, v3.0, and v3.1 to compare the severity breakdown of vulnerabilities in each scoring version over the past four year.

The most noticeable change that we saw in the update from v2 to v3 is that scores rose substantially, since a vulnerability that would have been rated as a 7.6 under CVSS v2 could quickly find itself with a 9.8 under CVSS v30. With CVSS v3.0, teams faced a higher number of high and critical severity vulnerabilities.

Still missing are the tools to prioritize and address them, or even fully understand the vulnerabilities' impact, on their project.

Severity Break-down: CVSSv2.0, vs. CVSSv3.0, vs. CVSSv3.1



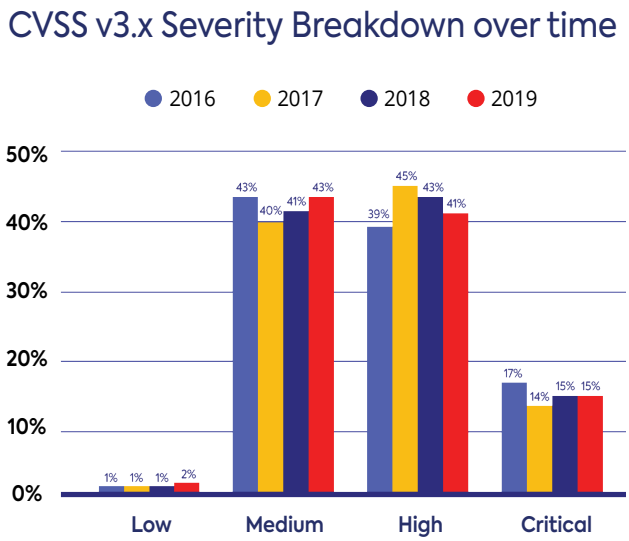
CVSS v2.0 Ratings	
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

CVSS v3.0 Ratings	
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Looking at the severity distribution of new CVSS v3.1 score, we can see that this is not a normal distribution, since 17% of the vulnerabilities are critical and only 2% are low.

While the community has been working to find an objective severity scoring standard to help users address the evolving security landscape, the standard is yet to be perfected. Additional factors that might cause this imbalance are the heightened focus on high and critical issues, and the fact that creating a CVE is a time-consuming effort that some prefer to avoid when it comes to lower-severity issues.

The question remains: how can we expect teams to prioritize vulnerabilities efficiently when over 55% are high-severity or critical?



Top Open Source Vulnerabilities Reported In 2019

Digging into the data on open source security vulnerabilities — their distribution across years, types, or severity scores, it's easy to forget that these issues occur in practically all of our favorite software projects — that just happen to be open source.

Wrapping up, let's take a look at the open source projects impacted by the open source security vulnerabilities that we've been studying so closely.

The most important takeaway from this list is that just because popular open source projects have vulnerabilities, that doesn't mean they are inherently insecure.

It only means that as a user of open source projects you need to be aware of the security risks and make sure to keep your open source dependencies up to date.

Open source components have become an integral part of our software projects. The open source vulnerabilities landscape might seem complex and challenging at first, but there are ways to gain visibility and control over the open source components that make up the products that we release.