WhiteSource

WHITESOURCE REPORT

WHAT ARE THE **MOST SECURE** PROGRAMMING LANGUAGES

CONTENTS

HOW DO THE TOP PROGRAMMING LANGUAGES MEASURE UP WHEN IT COMES TO SECURITY?

We all know that behind every developer is a beloved and often contentious programming language. In heated debates over which language is the best, the security card usually comes into play in support of one language or to discredit another. We decided to address this debate once and for all and put it to the test by taking a close look at seven of the most popular programming languages today to see which are the most secure.

Some developers and researchers claim that there is one language that is more or less secure than another, the truth is that there are many factors that go into choosing a programming language, and it's up to us to make sure, when we use it, that we are doing everything we can to ensure our software project's security.

The research below is based on WhiteSource's comprehensive database which aggregates information on open source vulnerabilities from multiple sources like the National Vulnerability Database (NVD), security advisories, GitHub issue trackers, and popular open source projects issue trackers.

Of the 200+ languages that the database covers, we focused on open source security vulnerabilities in the seven most widely used languages over the past ten years to find out which programming languages are most secure, which vulnerability types (CWEs) are most common in each language, and why.

This is what we found.

KEY INSIGHTS

1. Vulnerabilities in C amounted to 50% of all reported open source security vulnerabilities. This can be explained by the fact that it has been around the longest, has the highest volume of written code, and is the base of all the infrastructures that we use.

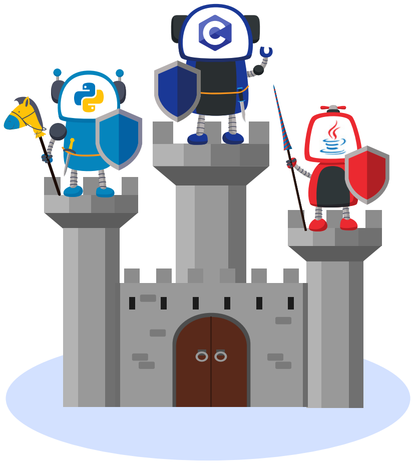
2. The most common CWE's across most programming languages are Cross-Site-Scripting (XSS); Input Validation; Permissions, Privileges, and Access Control; and Information Leak / Disclosure.

3. The use of automated tools and the trend of bug bounty programs have changed the game and led to a significant rise in reported vulnerabilities in 2017.

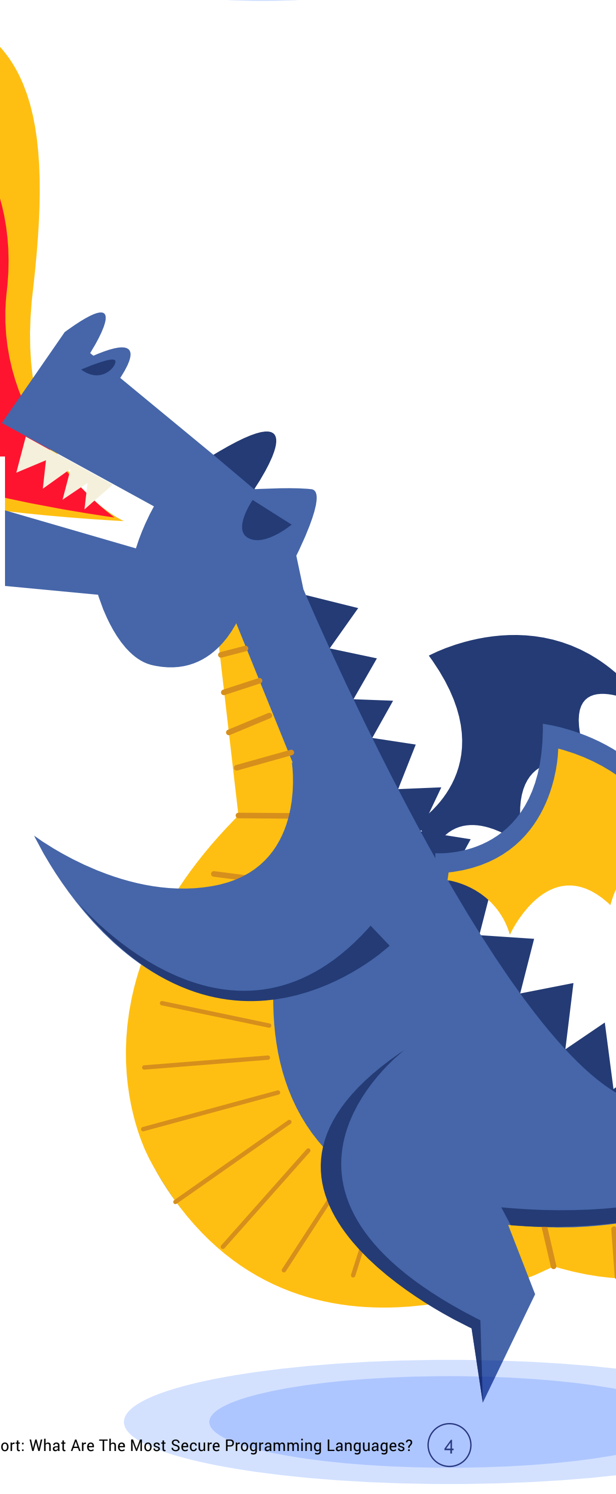
4. While we saw a spike in the number of reported security vulnerabilities over the past two years, the number of high severity vulnerabilities has decreased in most languages.

CHAPTER 1

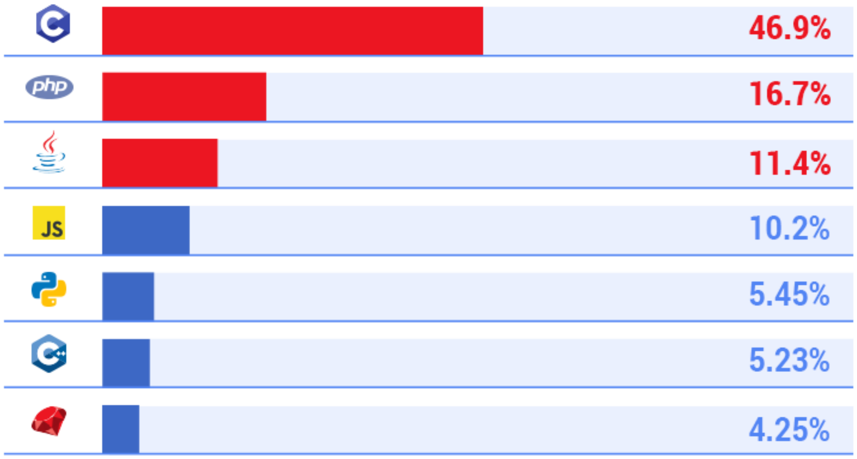
IS ONE LANGUAGE MORE SECURE THAN ANOTHER?



According to our knowledge base, C has the highest number of vulnerabilities out of all seven languages, with 50% of all reported vulnerabilities in the past 10 years.



Total Reported Open Source Vulnerabilities per Language



Before we delve further into the research, there are two considerations that we should take into account in assessing these projects. While on the face of the findings some might mistakenly assume that C is inherently more vulnerable, this is not the case.

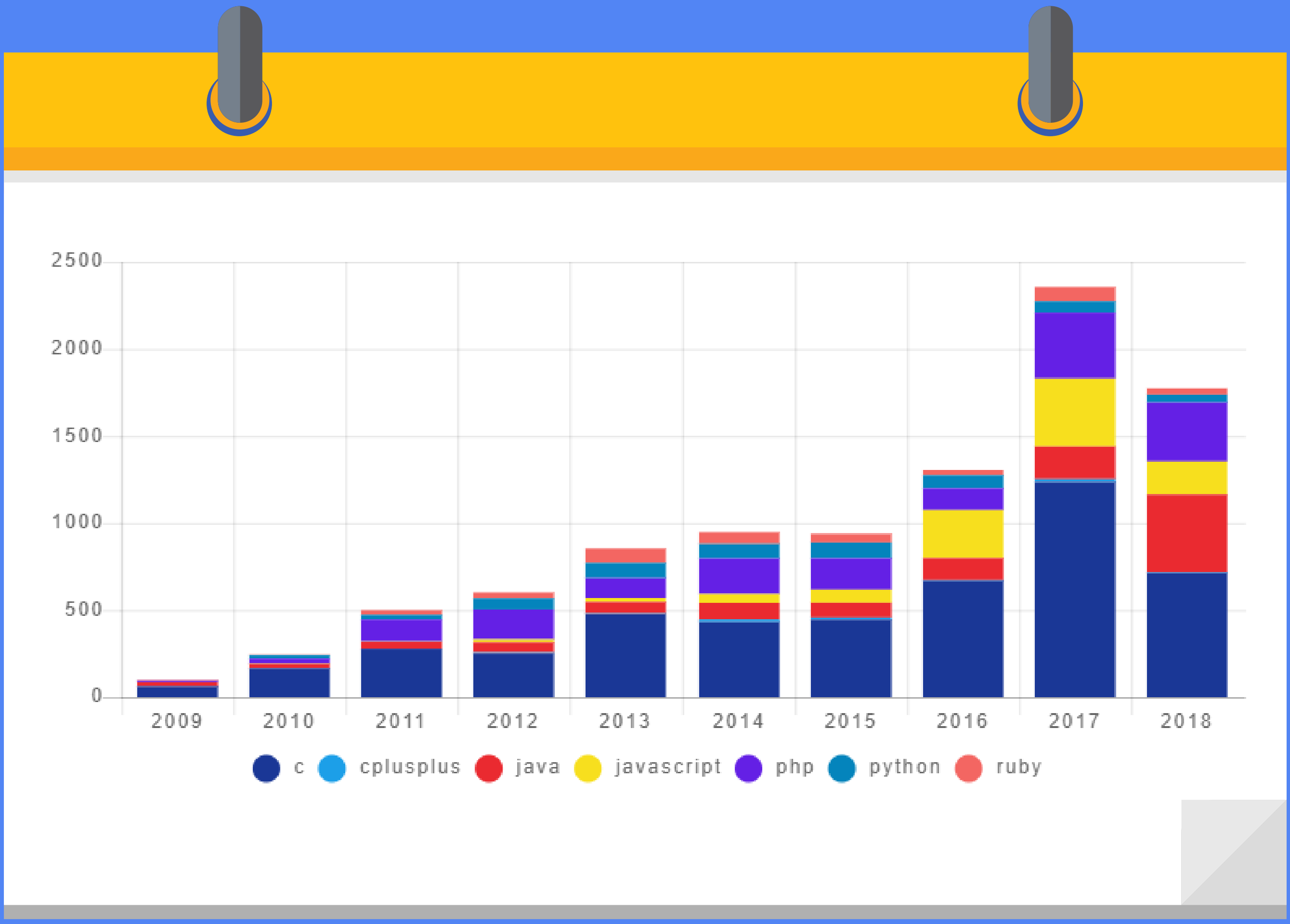
For starters, more code has been written than any other language, providing more opportunities for vulnerabilities to be discovered. The fact is that C has been in use for much longer than most other languages, and is behind the core of most of the products and platforms we use. As such, it is bound to have more known vulnerabilities than the rest.

VULNERABILITIES PER LANGUAGE OVER TIME

When we crunch the numbers and review the amount of reported open source vulnerabilities per programming language over time, what stands out is that there is no consistent trend for all languages apart from the fact that all languages saw a significant rise in the number of reported vulnerabilities in 2017.

While each language has had its own highs and lows, vulnerability-wise, over the past ten years, there are a few reasons behind the rise in vulnerabilities. Heightened awareness of security vulnerabilities in open source components, combined with the rise in the popularity of open source, have brought more focus to this open source security research. This attention has resulted in more issues being discovered. In addition, automated tools and the large investment in bug bounty programs have further contributed to the substantial increase in the number of reported open source issues.

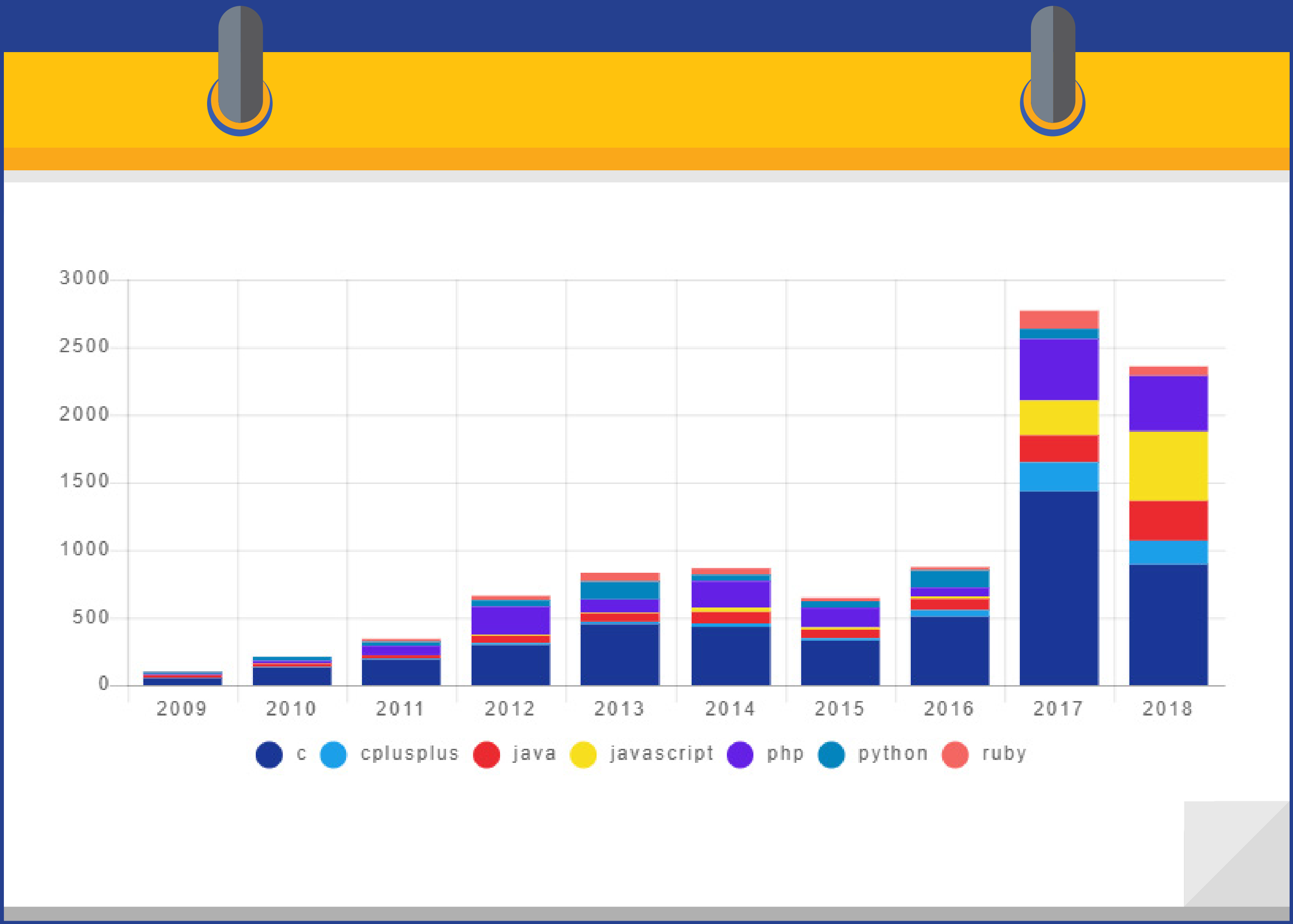
Vulnerabilities per Year, 2009-2018



HIGH SEVERITY VULNERABILITIES OVER TIME

When we went deeper into the vulnerabilities data and focused on vulnerabilities with a high severity (above 7 according to CVSS v2), we found that although there was a spike in the number of reported vulnerabilities in 2017, the percentage of critical vulnerabilities is declining in most of the languages we researched, excluding JavaScript and PHP.

The decrease in critical vulnerabilities might be explained by the concerted effort from security researchers to use automated tools to discover vulnerabilities in open source components. These tools most often are usually less capable of uncovering more complex and critical issues. While many of these tools are doing a good job of uncovering new vulnerabilities, many of the new security flaws discovered are not critical, and so we see a rise in the number of mostly medium vulnerabilities.



CHAPTER 2

DIFFERENT SECURITY VULNERABILITIES FOR DIFFERENT LANGUAGES (CWES)

We next chose to examine the types of vulnerabilities that were appearing in each language, to study another aspect of the threats to their security. To better understand their weak and strong points, we analyzed the types of CWEs that were found in each language over time.

When we examined the top three CWEs for each language, we noticed that two CWEs were at the top of the list for 70% of the languages: Cross-Site-Scripting (XSS) also known as CWE-79 and Input Validation also known as CWE-20.

Most Common	1	2	3
C	CWE-119 Buffer Errors	CWE-20 Input Validation	CWE-399 Resource Management Errors
php	CWE-79 Cross-Site Scripting (XSS)	CWE-89 SQL Injection	CWE-264 Permissions, Privileges, and Access Control
☕	CWE-200 Information Leak / Disclosure	CWE-20 Input Validation	CWE-79 Cross-Site Scripting (XSS)
JS	CWE-310 Cryptographic Issues	CWE-22 Path Traversal	CWE-79 Cross-Site Scripting (XSS)
C++	CWE-119 Buffer Errors	CWE-20 Input Validation	CWE-200 Information Leak / Disclosure
🐍	CWE-20 Input Validation	CWE-264 Permissions, Privileges, and Access Control	CWE-79 Cross-Site Scripting (XSS)
💎	CWE-79 Cross-Site Scripting (XSS)	CWE-264 Permissions, Privileges, and Access Control	CWE-20 Input Validation

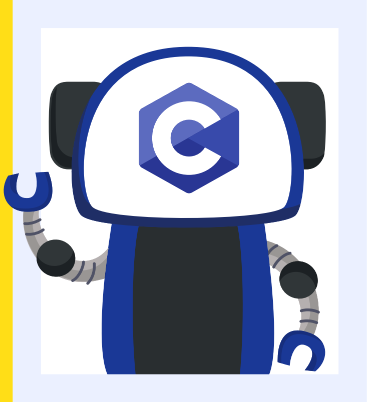
It is not surprising to see that most of the languages also share quite a few of their top ten CWEs, in addition to XSS and Input Validation, other CWEs are Information Leak/ Disclosure (CWE-200), Path Traversal (CWE-22), CWE-264 Permissions, Privileges, and Access Control, which was replaced in more recent years with Improper Access Control (CWE-284).

With the growing awareness of open source security vulnerabilities, we'd expect some types of vulnerabilities to disappear once they are discovered and addressed, but they don't. When we looked at the rise and fall of different CWEs in different languages over the years, we found that while some CWEs ebb and flow, on the most part, they refuse to go away:

Some vulnerabilities change names: CWE-264 (Permissions, Privileges, and Access Control) was common in 2012-2014, then decreased only to be replaced in the last 2-3 years by its more specific case - CWE-284 (Improper Access Control)

Some CWE's peak and then decrease: We see that Information Leak and Input Validation have risen and fallen over the years. This repeats in other CWEs as well. Over time, when vulnerabilities become more popular, frameworks solve them, or developers become more aware of them, and they decrease for a few years.

To gain insight into each language's security profile, we also dove deep into our data to find out which types of security flaws (CWEs) are most common for each language over time, and why.

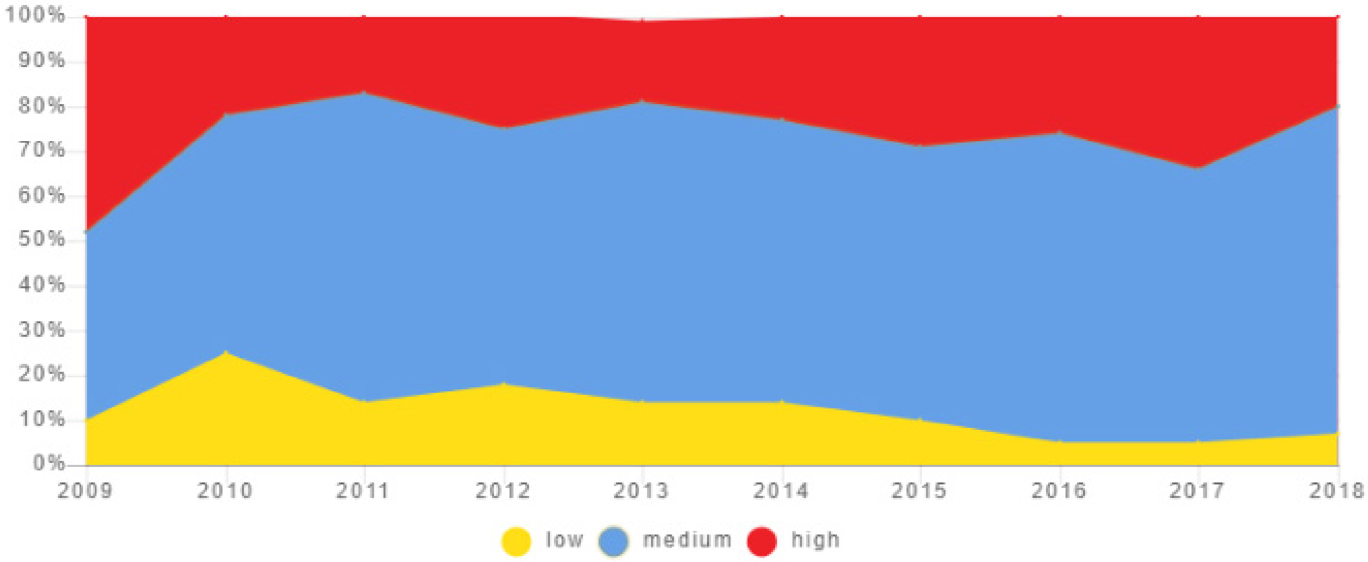


ID FOR C

Date of birth:	1972
Popular projects:	Linux, OpenSSL
High severity vulnerabilities over the past 5 years:	26% on average, with a significant spike in 2017

C is by far the language with the highest number of reported vulnerabilities of the bunch. Vulnerabilities in C account for over 50% of all reported open source vulnerabilities since 2009, and although we see the number of vulnerabilities rise and fall to some extent over those years, the amount of C vulnerabilities each year far surpasses the rest of the programming languages.

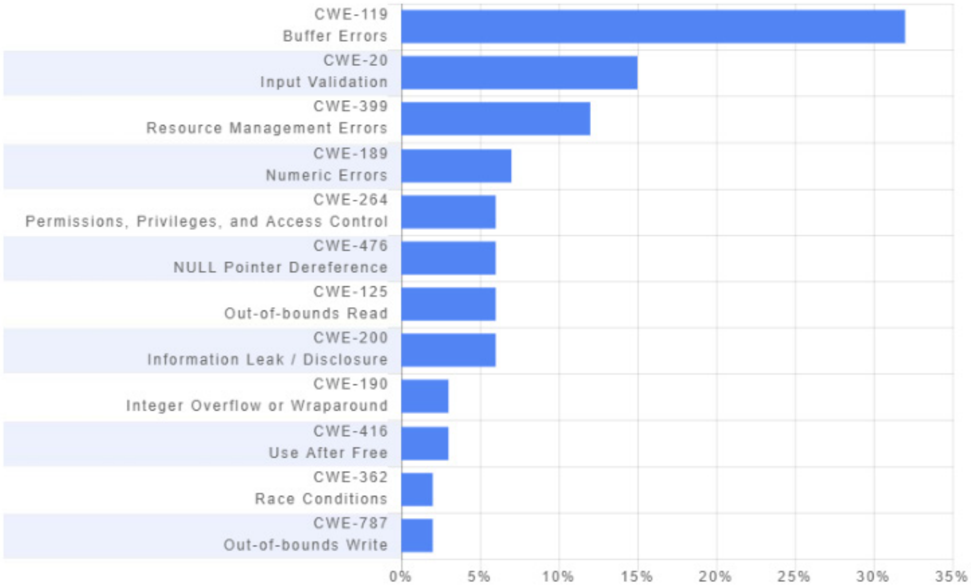
While it's not where the cool kids go to create web applications, C is behind what most of us do. It powers the Linux Kernel, and a variety of other projects that many of us might be surprised by how common they are, like FFmpeg, curl, and ImageMagick. C projects combined boast a huge open source development community — possibly the largest open source community out there.



C Security Vulnerabilities: Top CWEs

Buffer Errors (CWE-119) are the most common security vulnerability in C by a wide margin, singular to C and C++'s vulnerability profile. This is understandable since most of the CWEs that are common in other languages are related to web and web services issues, which are not relevant in C.

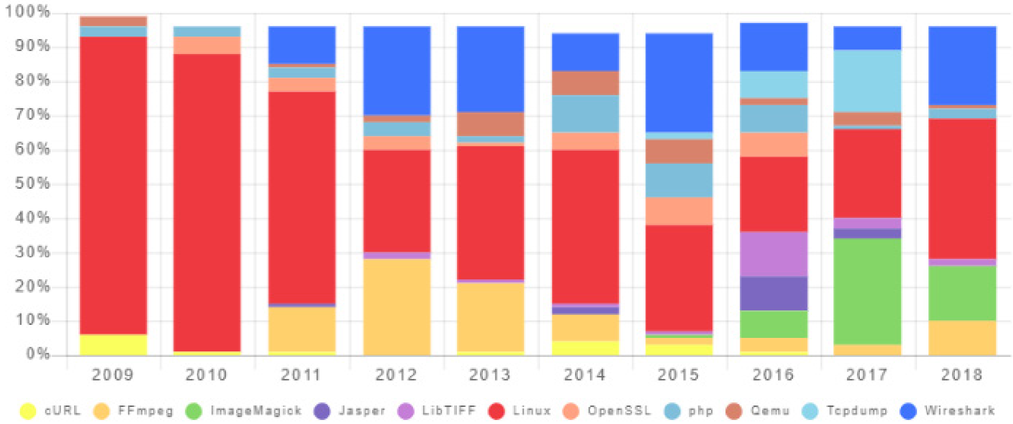
It's important to note that this group of vulnerabilities (sometimes called memory corruption) can often have critical consequences.

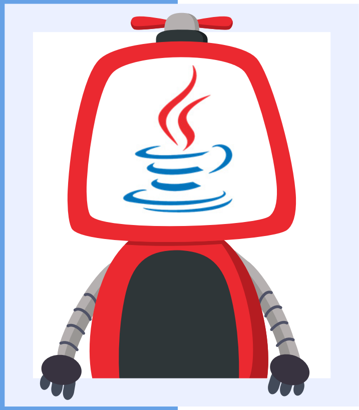


C Security Vulnerabilities per Project

While most languages are used to build hundreds, if not thousands of packages, it's easy to track the open source projects that are built on C. This allows us to learn exactly which open source libraries experienced growth in reported known vulnerabilities, following the rise in vulnerabilities over the years.

We can see that Linux vulnerabilities have nearly always accounted for a high percentage of vulnerabilities in C. Since a notorious vulnerability lovingly named ImageTragic was discovered in mid-2016, security researchers must have doubled-down to ensure that it's safe to use, resulting in a huge spike of discovered ImageMagic vulnerabilities in 2017.





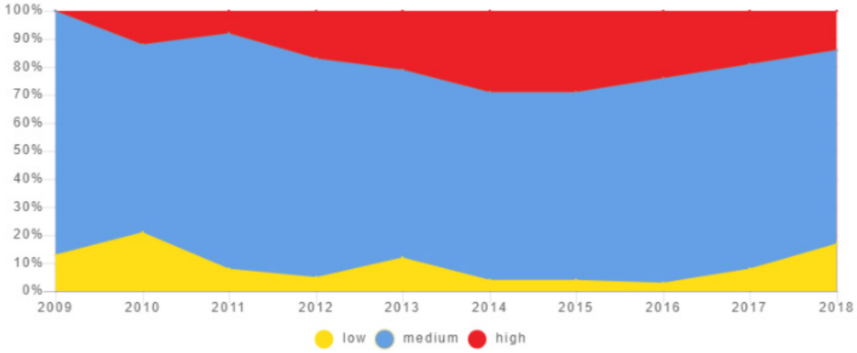
ID FOR JAVA

Date of birth:	1995
Popular projects:	Android SDK, Spring Framework
High severity vulnerabilities over the past 5 years:	19% on average, but consistently declining since 2015

JAVA Security Vulnerabilities: per Severity

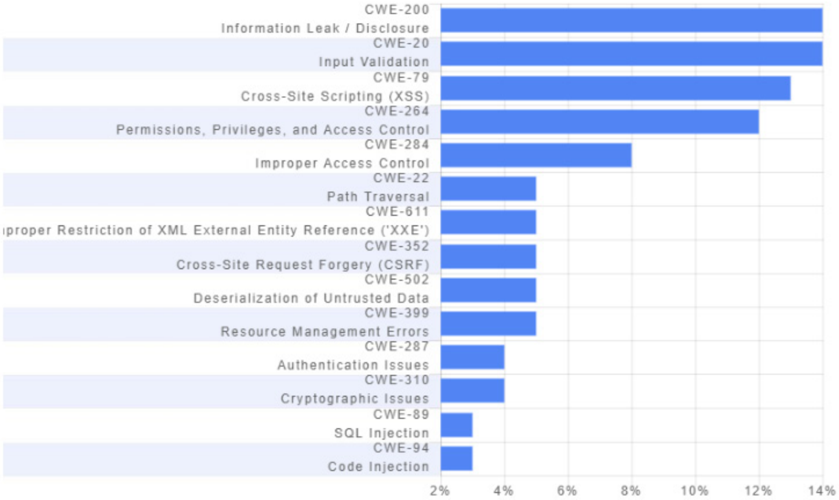
Java vulnerabilities have been consistently rising since 2016. While for most languages in this report the numbers went down this year, Java is the only language that saw a rise in open source vulnerabilities this past year. Surprisingly perhaps, Java vulnerabilities nearly doubled in 2018 as compared to 2017.

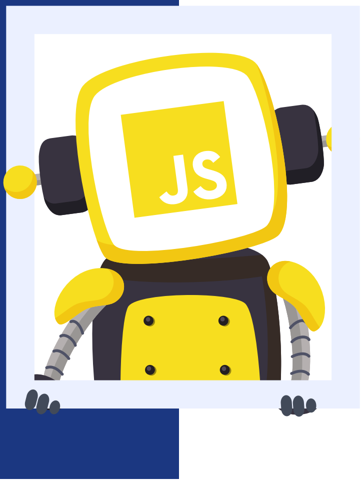
Java mid severity vulnerabilities are pretty steady around the 75% mark, but we can see that the % of high severity vulnerabilities are increasing the past two years, reaching 20% in 2018.



Java Security Vulnerabilities: Top CWEs

Java shares Python's top four CWEs: Deserialization issues (CWE-502) is one vulnerability type that while not the most prominent, is unique to this language and saw a rise in 2017. Deserialization issues don't appear in PHP, Ruby, or Python. It may be worth asking if researchers should start looking for deserialization issues in those languages too, or is serialization inherently more secure in the other languages?





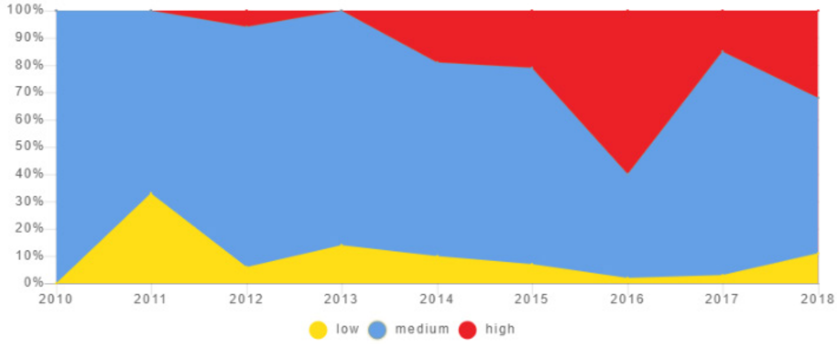
ID FOR JAVA SCRIPT

Date of birth:	1995
Popular projects:	Node.js, React
High severity vulnerabilities over the past 5 years:	31% on average, but has one of the most inconsistent trends

JAVA Script Security Vulnerabilities: per Severity

JavaScript, arguably the most popular language, is one of the only languages that saw a continuous rise in the number of vulnerabilities in the past ten years. In 2017 the number of reported vulnerabilities was 16 times higher than in 2016 and continued to rise in 2018. Although the number of JS vulnerabilities increased in 2018 by over 50%, the number of high severity vulnerabilities stayed the same in 2017 and 2018.

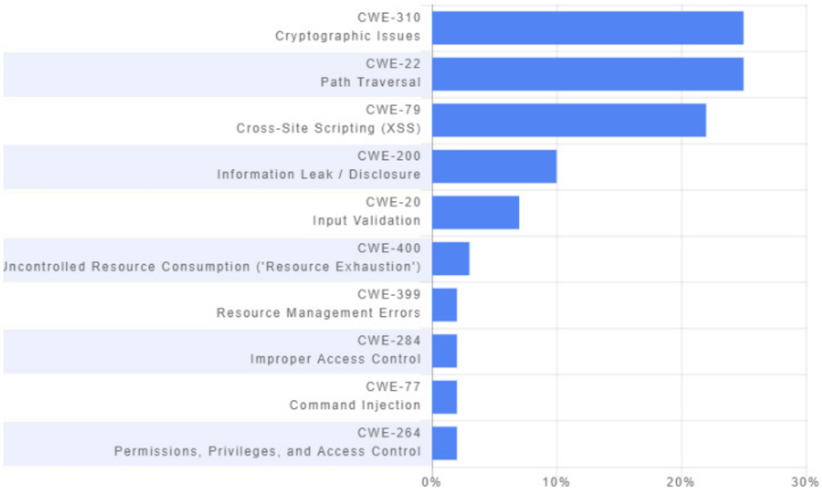
The rise in the number of known vulnerabilities may be attributed to its rising popularity, along with the fact that JS has become popular as a language for backend in recent years.



JavaScript Security Vulnerabilities: Top CWEs

JavaScript's top two most common CWEs - Cryptographic Issues (CWE-310), and Path Traversal (CWE-22) are anomalies since these issues are unique to JS, they are much more common there than in any other language we researched. Many of them come from a few researchers and are vulnerabilities in unpopular or even dead packages. We take them to be unnatural CVEs.

When we looked into the NPM packages, we found that while 61% of the JS vulnerabilities are path traversal and crypto, 70% of those packages are barely used, maintained or supported, and had less than 2000 downloads in 2018.

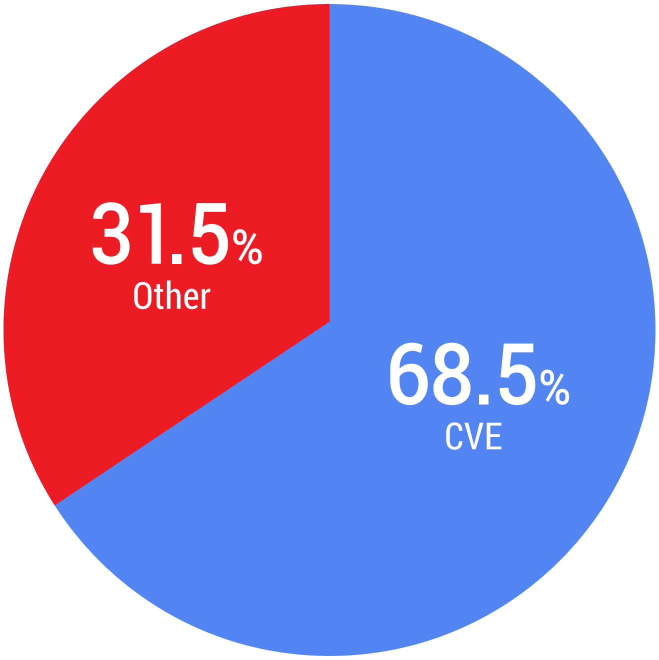


One of the reasons for the inflation of these particular CWE's, in packages that are scarcely maintained or downloaded, is the adoption of new automated tools to discover certain types of CWEs. When we look at the years these vulnerabilities spiked, we see that nearly all of the Cryptographic Issues (CWE-310) were found in 2016, and the vast majority of the Path Traversal issues (CWE-22) were found in 2017. It is very likely that the ease of using automation to locate CWE-310 and CWE-22 vulnerabilities in JavaScript are behind the unusually high number of these types of CWEs.

This proves once again that just looking at the number of vulnerabilities isn't enough. If we really want to understand how vulnerable a programming language or project is and what its weak points are, then we need more than just a quantifiable analysis.

JavaScript and the Community

Another interesting trend that we saw when we studied the JS open source vulnerabilities data is that over 30% of JavaScript vulnerabilities were disclosed on community platforms outside of the NVD.



Because of the decentralized nature of the open source community, information about open source vulnerabilities is scattered across hundreds of security advisories, open source project issue trackers and forums. This information is sometimes under-the-radar, with varying levels of credibility. This makes verifying that an open source component is secure and complies with your company's policies has become a very complex process, as the JS vulnerabilities data clearly demonstrates.

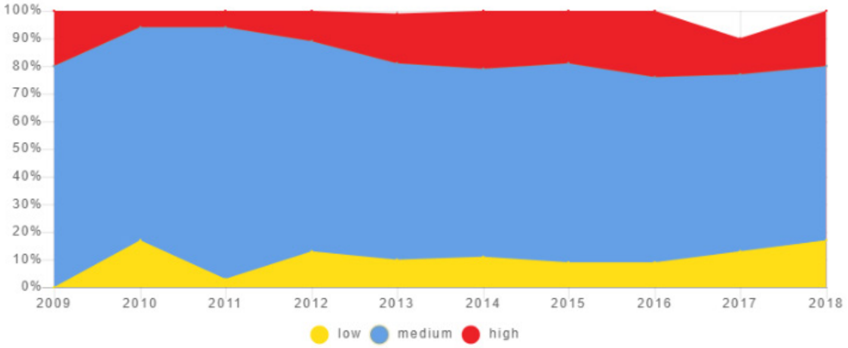
ID FOR PHP

Date of birth:	1995
Popular projects:	WordPress, Drupal, Moodle
High severity vulnerabilities over the past 5 years:	16% on average, consistent excluding a sharp decline in 2017

PHP Security Vulnerabilities: per Severity

PHP is very much a web-facing language and a go-to for young developers because of its low learning curve.

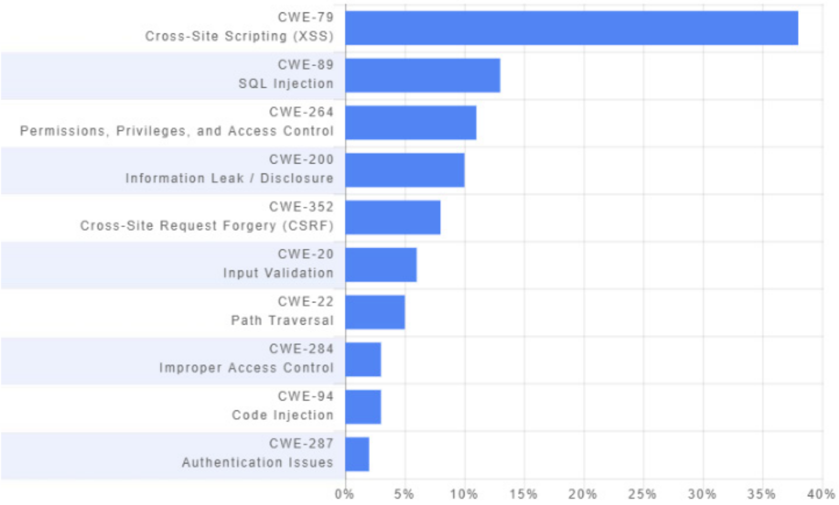
PHP's popularity has been in decline for the past few years. Throughout its time, the number of vulnerabilities has been the second highest of all the languages that we've included in this list, rising and falling in cycles since 2009, with a sharp increase in vulnerabilities in 2017 like the other projects we've studied.



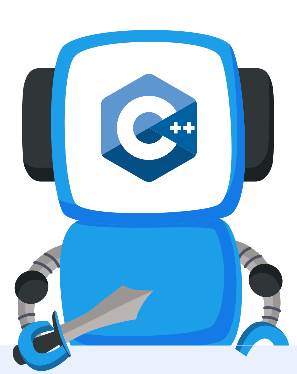
PHP Security Vulnerabilities: Top CWEs

PHP shares three of Ruby's top 4 most common CWEs, with XSS at the top. However, PHP is the only language with SQL Injection (CWE-89) vulnerabilities featured so prominently at the top of the list.

Many security experts would expect SQL injection vulnerabilities to be a thing of the past, however, this is not the case for PHP, where the SQL injection vulnerabilities have been common for years. There are PHP haters out there that say that some of the design flows and bad practices that are built into the language make it hard to write secure code and maintain a high level of secure coding standards. These issues are continuing to rise, and the number of SQL Injection vulnerabilities has been particularly high in 2017 and 2018.



CWE-79: Cross Site Scripting is the most common vulnerability here, and XSS vulnerabilities account for nearly 40% of the PHP CWEs since 2009.

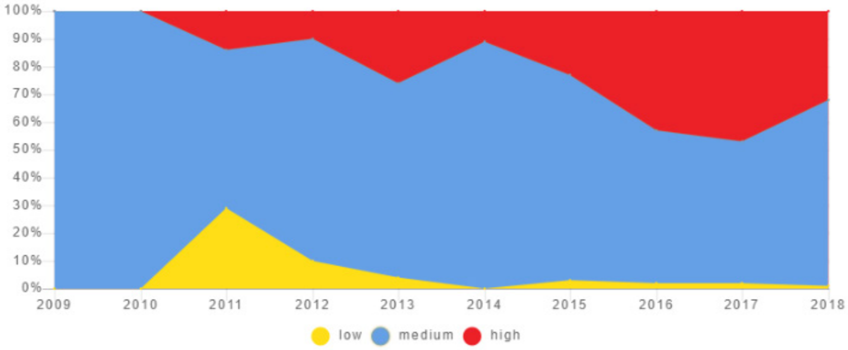


ID FOR C++

Date of birth:	1983
Popular projects:	TensorFlow, Electron
High severity vulnerabilities over the past 5 years:	36% on average, the highest % of all languages

C++ Security Vulnerabilities: per Severity

C++ and C suffer from the same types of CWEs. Since these languages are less popular in the development of web applications, the CWEs that plague the other languages are less common here.

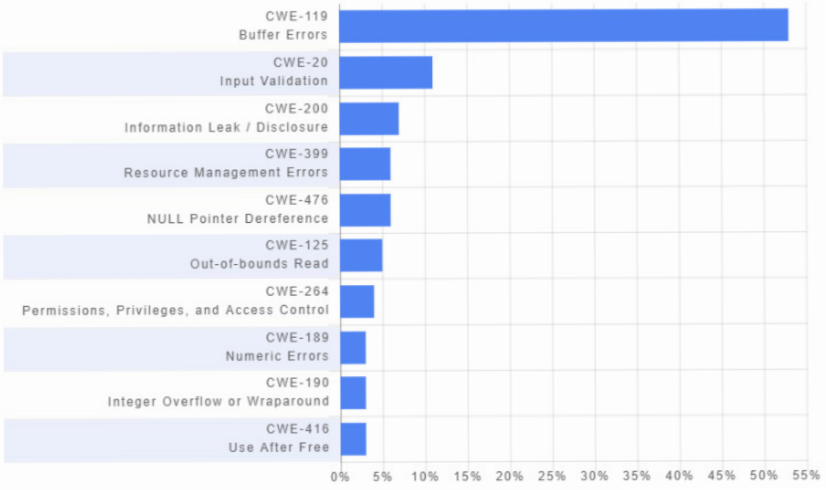


C++ Security Vulnerabilities: Top CWEs

Buffer Errors (CWE-119), have been very common in C for years, but C++ only recently started catching up to its compatriot with an extremely sharp spike in Buffer Error issues reported in 2017.

While Buffer Error vulnerabilities are the most prominent for C++, especially following the recent rise in discovered CWE-119 issues, Input Validation issues (CWE-20), which feature prominently in all of the languages that we researched, are the second most common vulnerability in C++ and have been increasing from 2016 to today.

This rise might also mean that Input Validation issues are at the top of the list for what security researchers are looking for today.



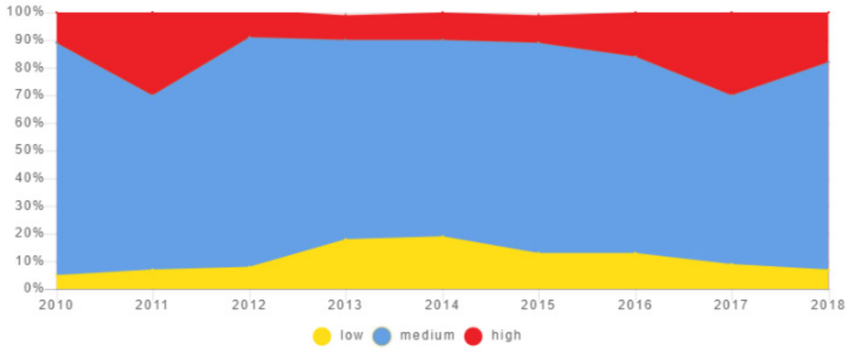
ID FOR PYTHON

Date of birth:	1991
Popular projects:	Django, Ansible
High severity vulnerabilities over the past 5 years:	15% on average, the lowest of the bunch

Python Security Vulnerabilities: per Severity

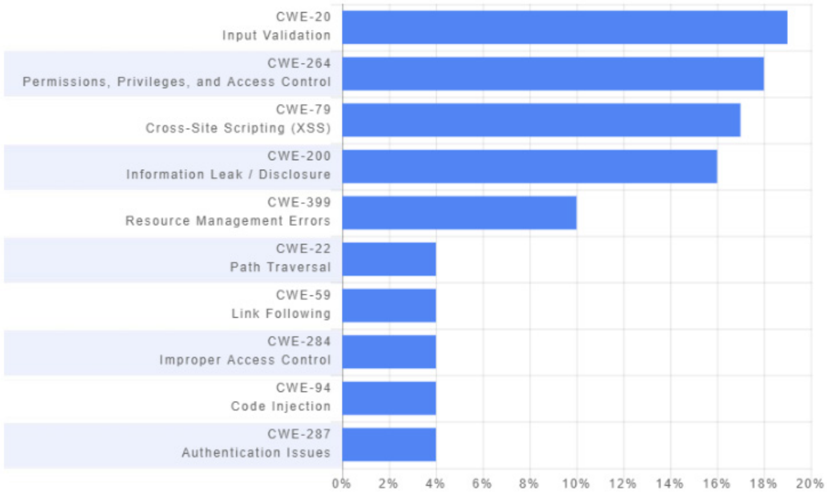
Over the past few years, Python has been gaining more popularity, and its security profile vulnerabilities-wise has been fairly consistent for quite some time. As opposed to most other languages that saw a rise in vulnerabilities in 2017 and a decrease in 2018, Python vulnerabilities reached a peak in 2015 and have been decreasing consistently since then.

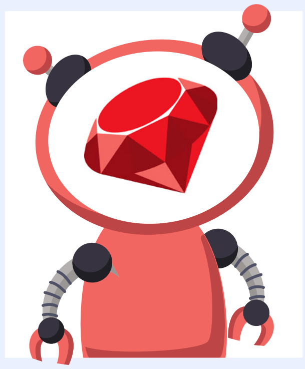
When we look at the breakdown of vulnerabilities over time, we can see that throughout the past 10 years, Python has suffered from a relatively small percentage of high severity vulnerabilities, up until 2017.



Python Security Vulnerabilities: Top CWEs

There are 4 types of vulnerabilities that dominate Python's CWEs list: Input Validation (CWE-20), Permissions, Privileges, and Access Control (CWE-264), Cross-Site Scripting (XSS) (CWE-79), and Information Leak / Disclosure (CWE-200). These vulnerabilities were prominent in most of the languages we looked at, and are dentical to Java's top five most common CWEs.



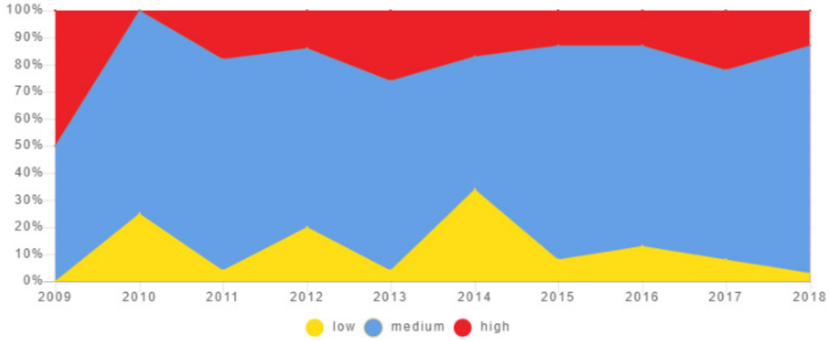


ID FOR RUBY

Date of birth:	1995
Popular projects:	Rails, Homebrew
High severity vulnerabilities over the past 5 years:	19% on average, pretty stable aside from a peak in 2017

Ruby Security Vulnerabilities: per Severity

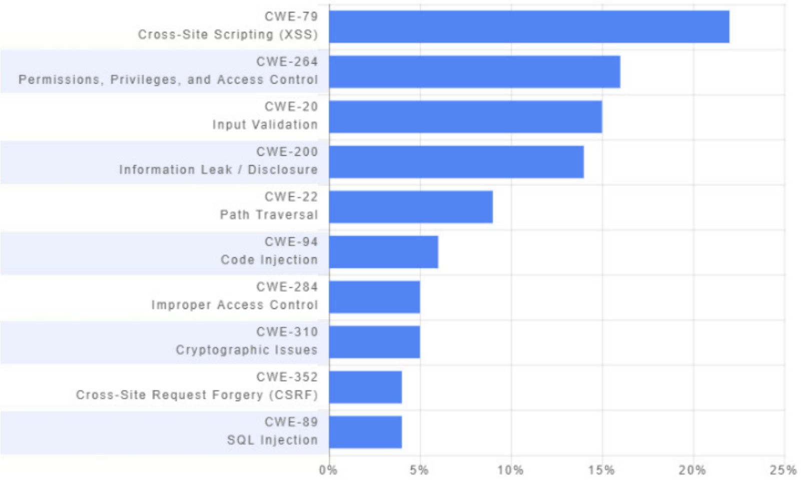
Of all seven languages, Ruby has the least amount of security vulnerabilities. The number of open source vulnerabilities in Ruby has repeatedly risen and fallen over the past 10 years, while its popularity seems to be on a steady decrease since 2015. The rise-and-fall trend continued throughout 2016-2018.



Ruby Security Vulnerabilities: Top CWEs

In terms of CWEs, XSS vulnerabilities, which are common in most languages, are the most common CWE in Ruby. The other top CWE's in discovered in Ruby, CWE-20 (Input Validation) and CWE-200 (Information Leak / Disclosure) both peaked in 2013-2014, and haven't been very prominent since then — meaning that most developers have learned how to mitigate them. CWE-264 (Permissions, Privileges, and Access Control), also a very prominent issue in 2012-2013, has decreased since then, but CWE-284 (Improper Access Control), its younger sibling, started appearing in Ruby in 2014.

Ruby is also the only language aside from PHP that has a relatively high number of SQL Code Injection vulnerabilities, although they are nowhere near the numbers in PHP and have pretty much died down since 2015. This is most probably another issue that developers have figured out how to avoid.



THINK WE'RE EXAGGERATING? **TRY US OUT!** Sign up for a free trial and be amazed by the ease and accuracy of the WhiteSource solution. **www.whitesourcesoftware.com**