The 2022 State of Open Source Vulnerabilities

FOSSA

Content

Introduction

2021 was an eventful year for software supply chain security.

It started amid the ongoing fallout from the SolarWinds supply chain hack. Then, in May, the Biden Administration published its landmark Executive Order on Improving America's Cybersecurity, which included several provisions intended to bolster the software supply chain.

Finally, in December, the security world was shaken when the Log4J remote code execution vulnerability wreaked havoc on applications everywhere.

The Log4J incident also shined a light on an important piece of the broader software supply chain security conversation: open source vulnerability management.

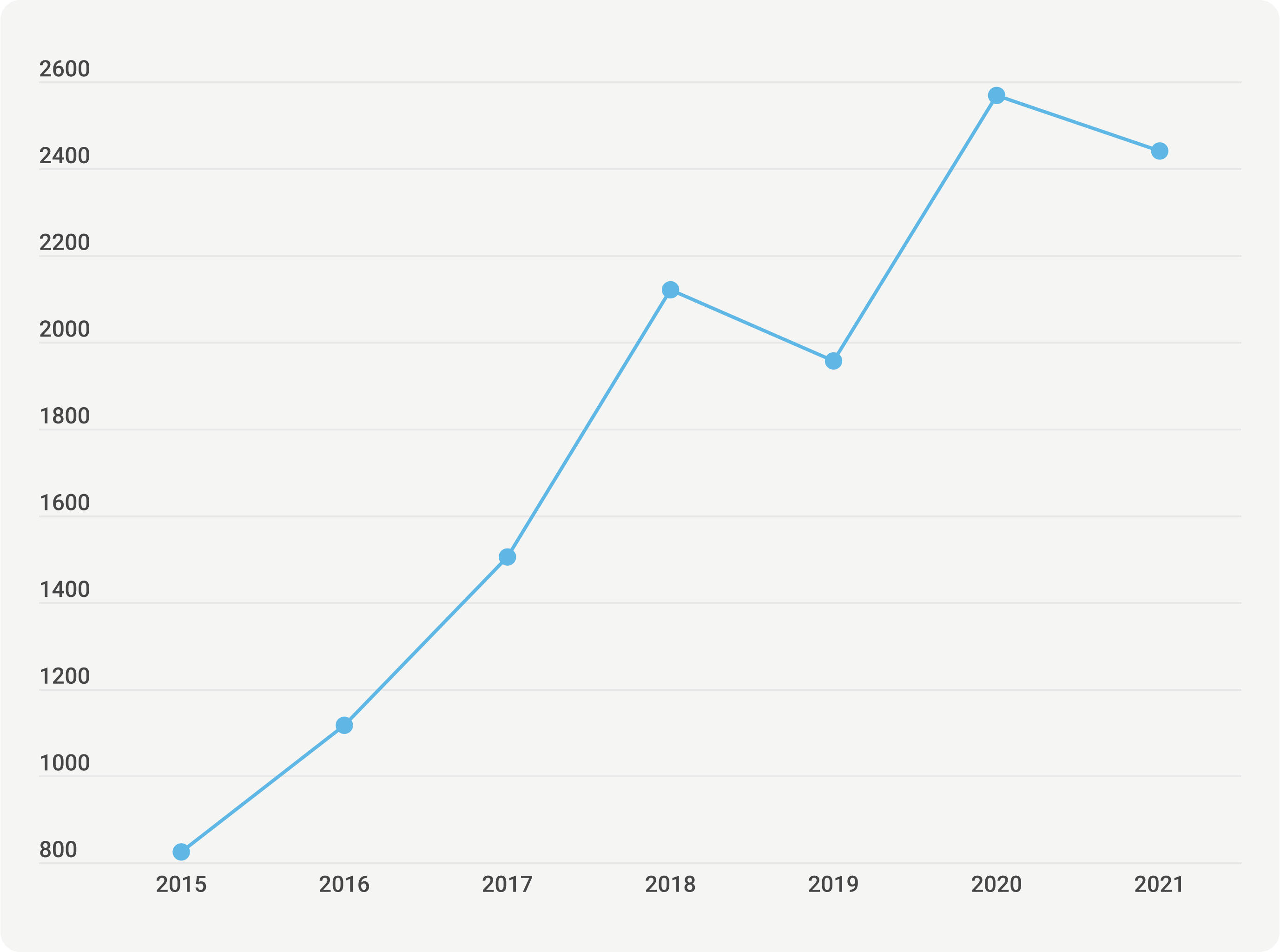
Log4J — a popular java open source logging library used in virtually all Java applications across the world — is just one of countless open source components that power modern software development. And, as ever-more sophisticated bad actors look for ways to gain unauthorized access to valuable systems and data, open source has become a significant attack vector.

To better understand the current open source threat landscape, we recently examined the FOSSA Vulnerability Database — sourced from multiple vulnerability feeds as well as our own research team — to gather insights into trends in open source vulnerabilities. The 2022 FOSSA Open Source Vulnerability Report also includes new research into vulnerabilities in Linux distributions.

Specifically, the report offers insight into areas like:

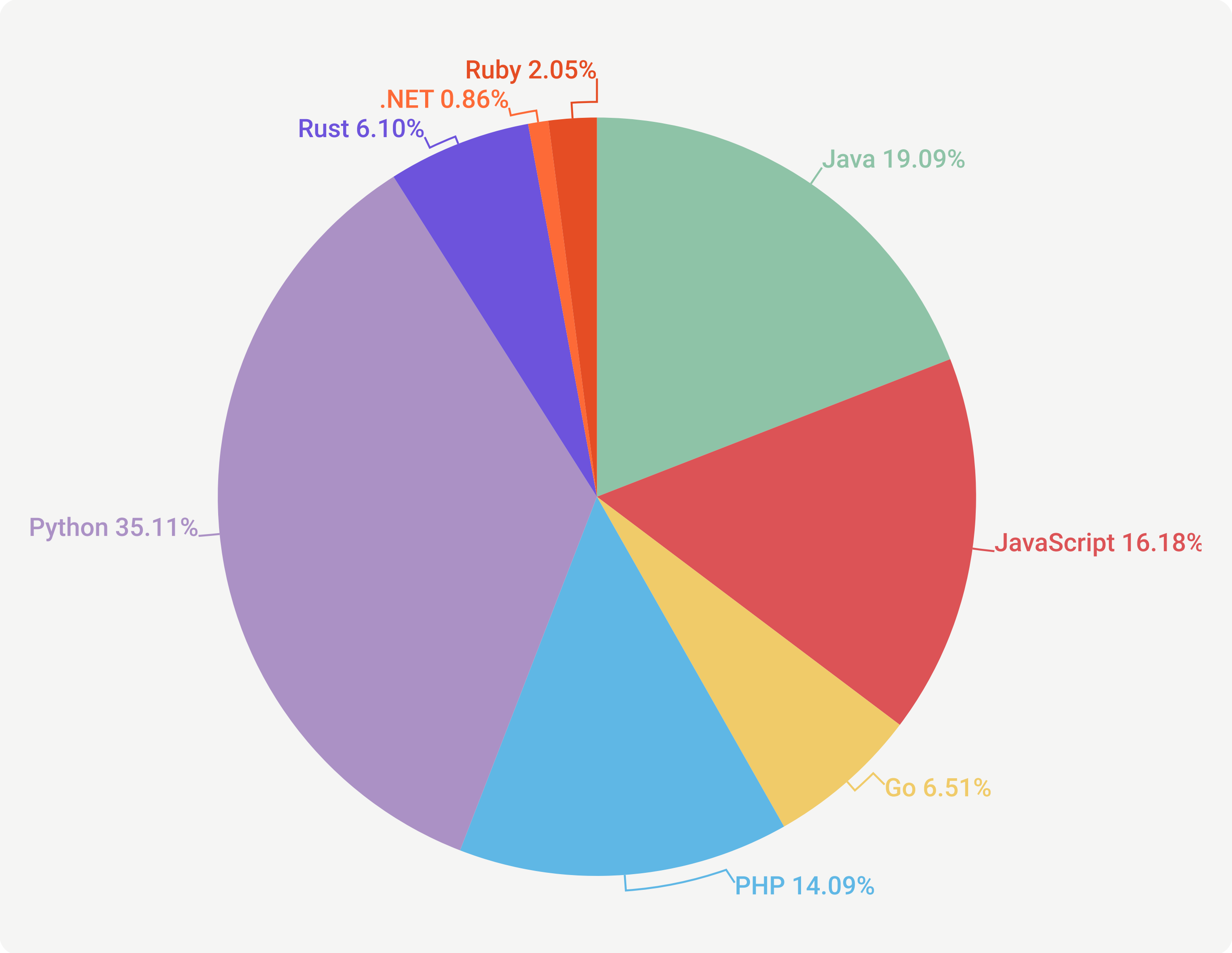
- ✓ The distribution of vulnerabilities across popular languages
- ✓ Longitudinal trends in vulnerabilities over the past five years
- ✓ The most prevalent CWEs in each language
- ✓ Vulnerabilities in popular Linux distributions

High-Level Trends for 2021



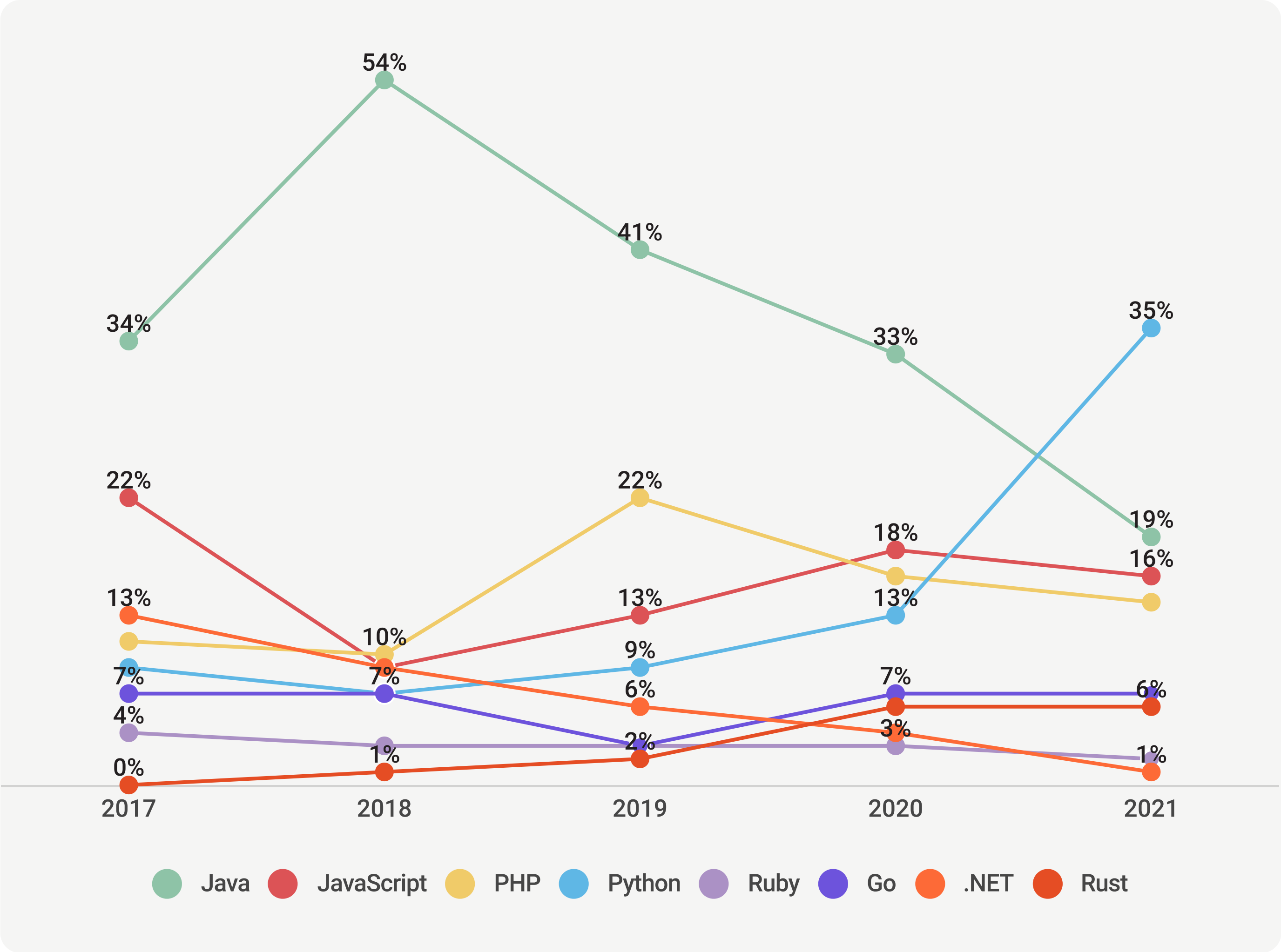
The number of new vulnerabilities discovered in 2021 was slightly lower than in 2020. However, the drop wasn't significant — and it was just observed over one year — so it is too early to draw any conclusions.

2021 Vulnerabilities by Ecosystem



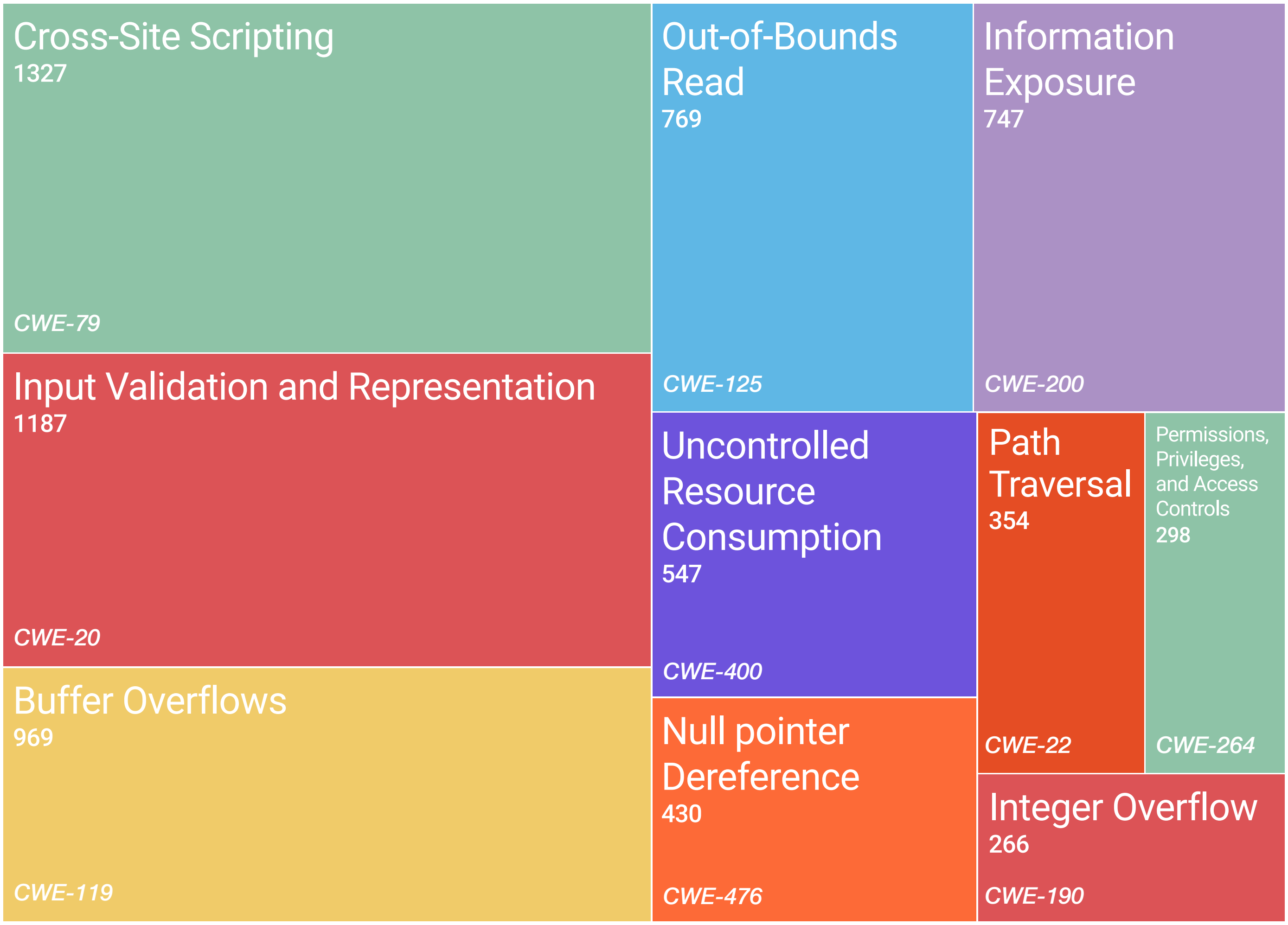
In 2021, Python had the highest number of new vulnerabilities discovered, followed by Java and JavaScript. These are some of the most commonly used languages, so it's not surprising that they also had a high number of vulnerabilities.

Yearly Distribution of Vulnerabilities by Ecosystem



As we see in the chart above, 2021 was the first year where Python had the biggest share of new vulnerabilities — in previous years, Java topped that list. It remains to be seen whether this was the start of a trend or more of an outlier. Go and Rust maintained their increased share of vulnerabilities since 2020, reflecting their increased prevalence.

Top 10 Most Commonly Found CWEs



A majority of the vulnerabilities discovered in 2021 were not based on new attack vectors. Instead, they were instances of old and common CWEs — such as new variants of categories like CWE-79: Cross-Site Scripting. CWE-79 is still the leading category of vulnerability overall across the past couple of years.

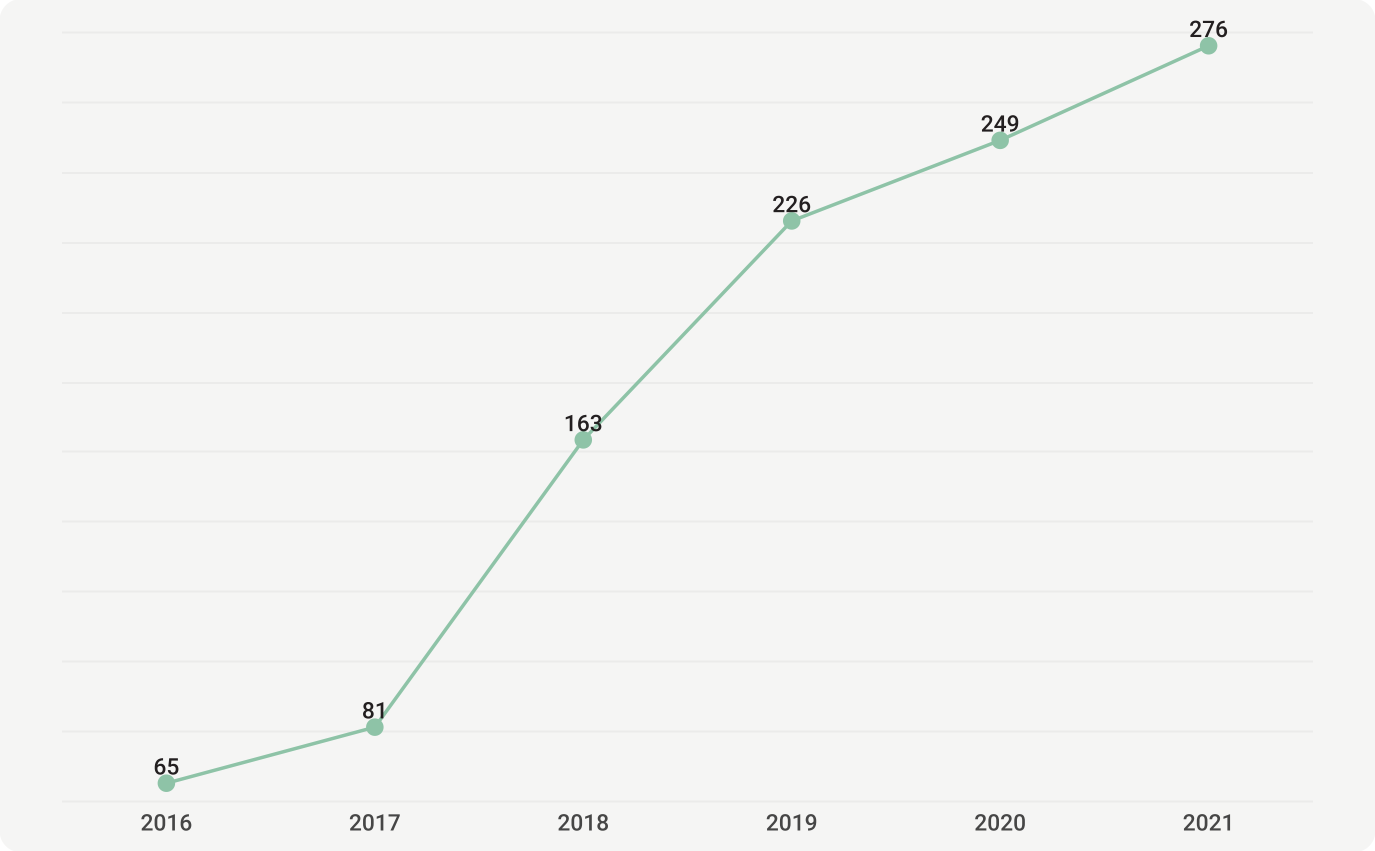
CWE-79: Cross-Site Scripting

CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-Site Scripting')

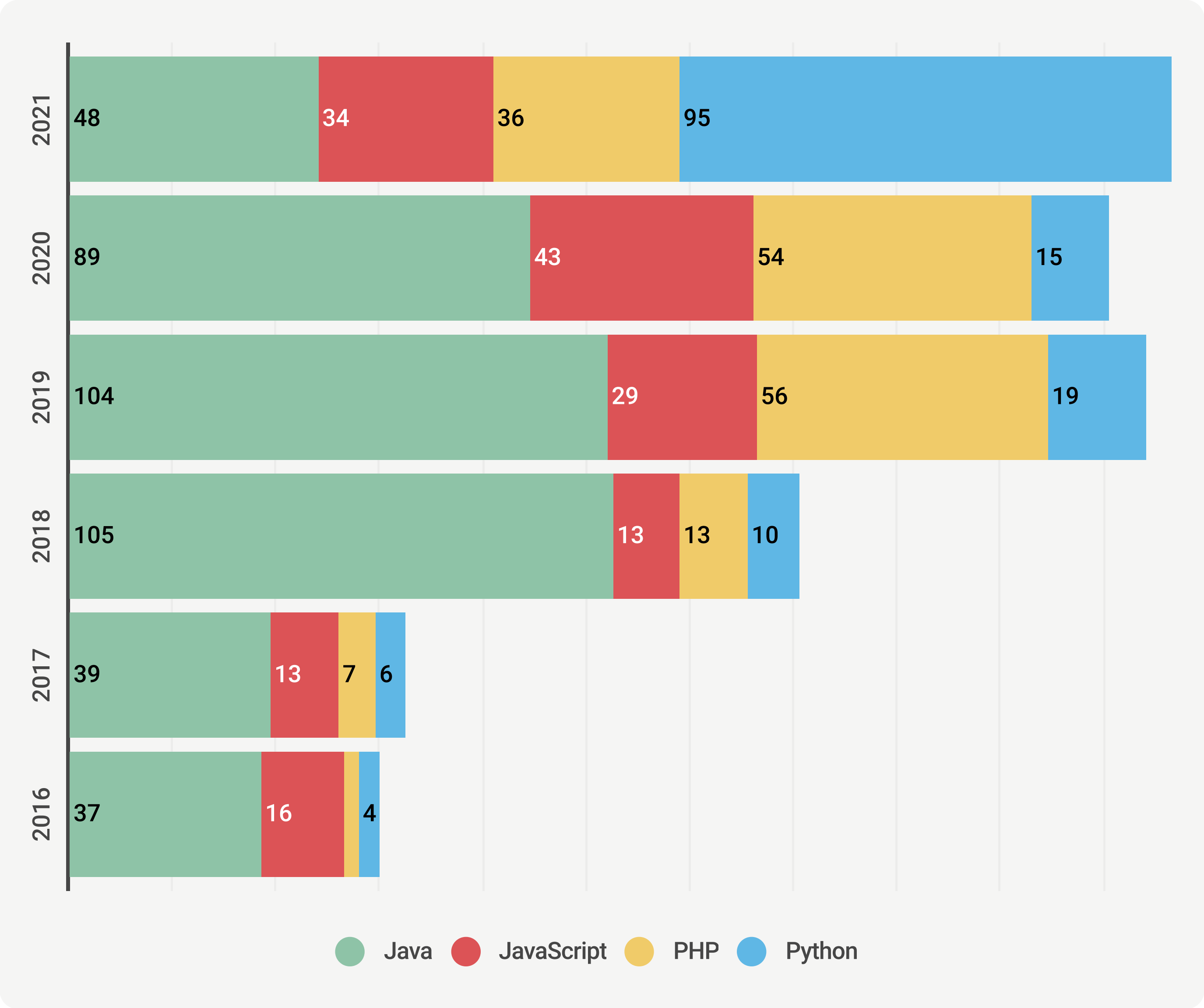
The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

From MITRE

CWE-79: Vulnerabilities Discovered by Year



CWE-79: Vulnerabilities by Ecosystem

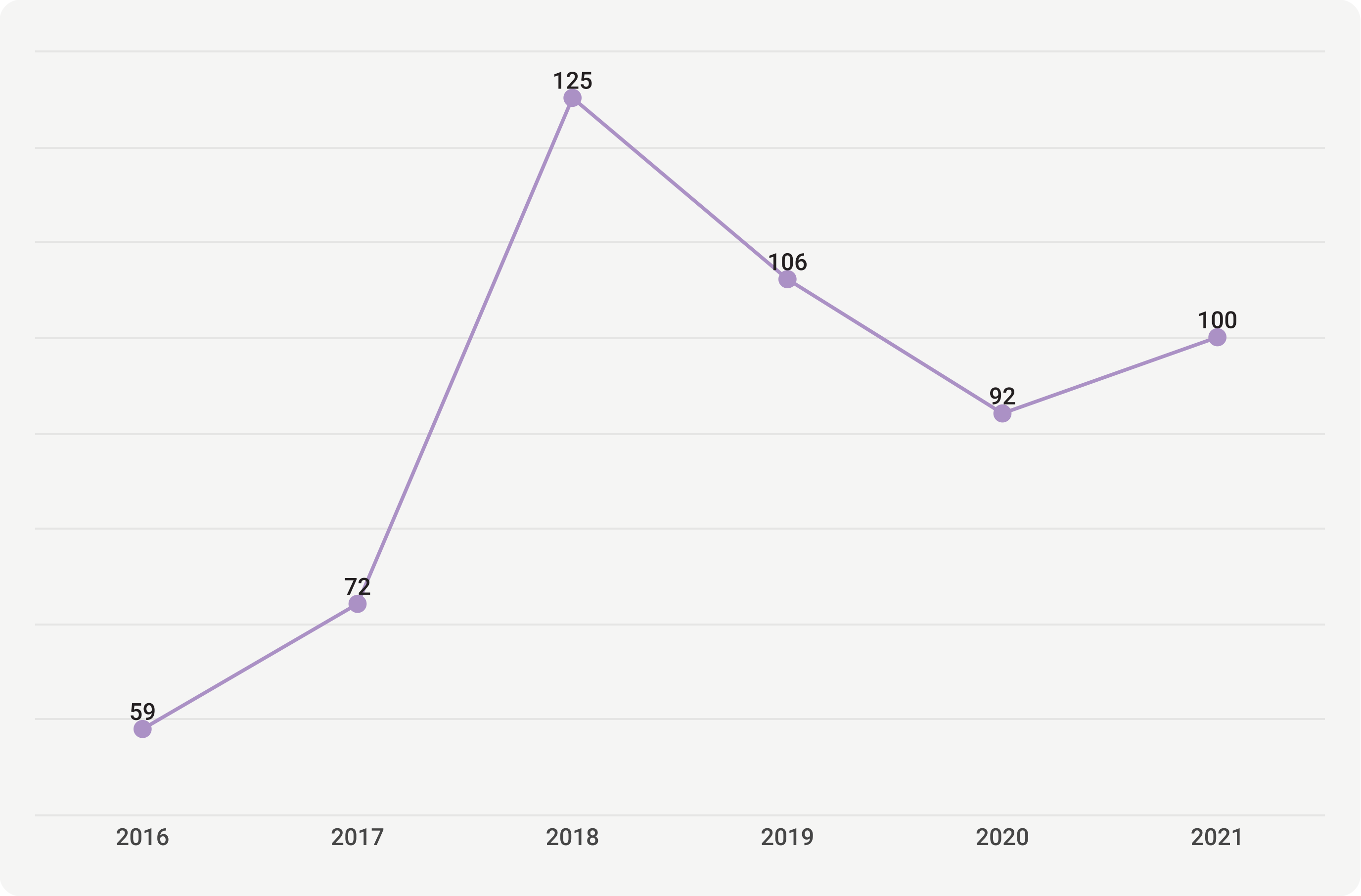


There was a drop in the number of new cross-site scripting vulnerabilities discovered across the Java, JavaScript, and PHP ecosystems in 2021. However, Python saw a 500-plus percent gain in cross-site scripting vulnerabilities from 2020 to 2021, which was the largest increase of any programming language.

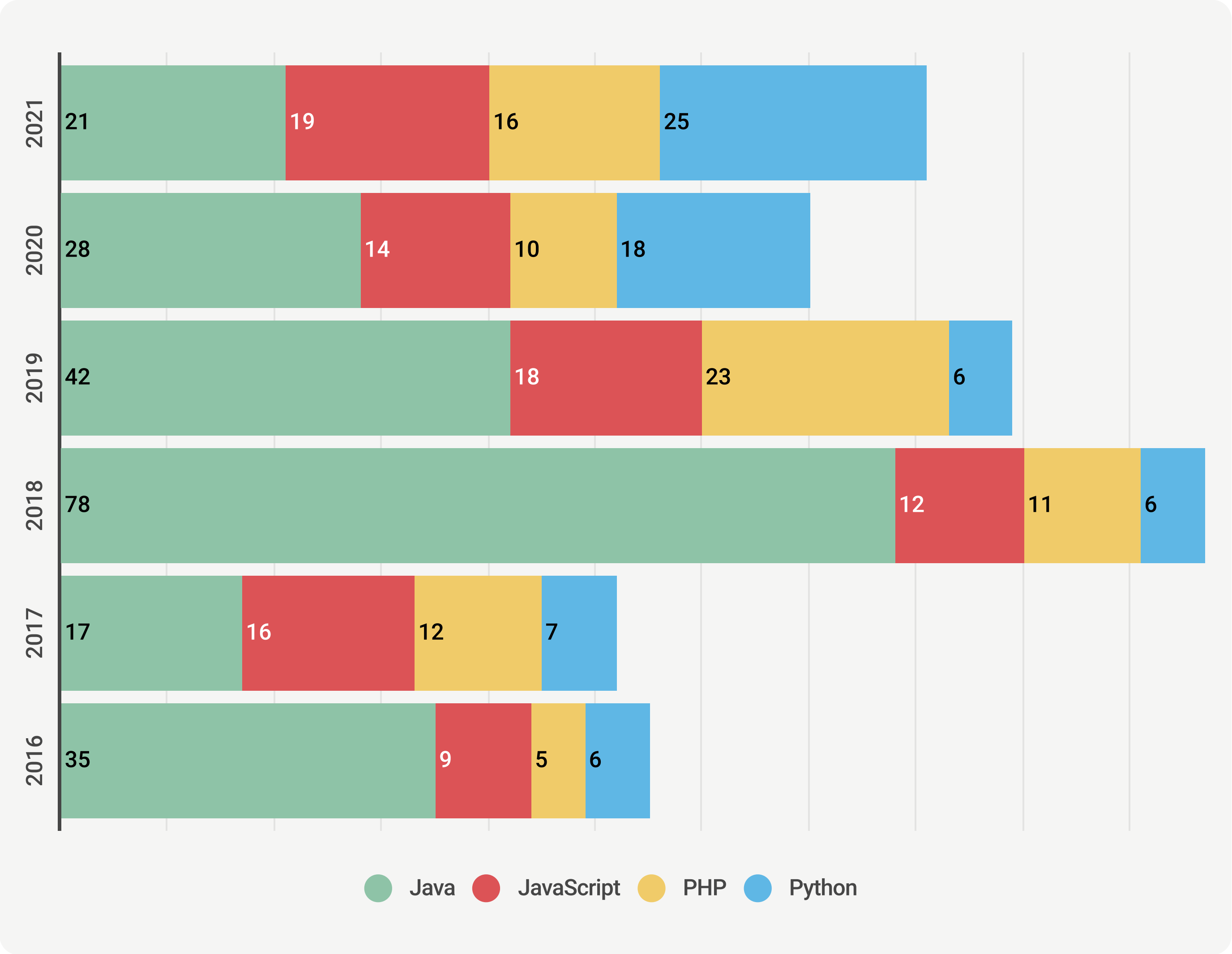
CWE-20: Improper Input Validation

> ***CWE-20: Improper Input Validation***
>
> *The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.*
>
> *From MITRE*

CWE-20: Vulnerabilities Discovered by Year



CWE-20: Vulnerabilities by Ecosystem

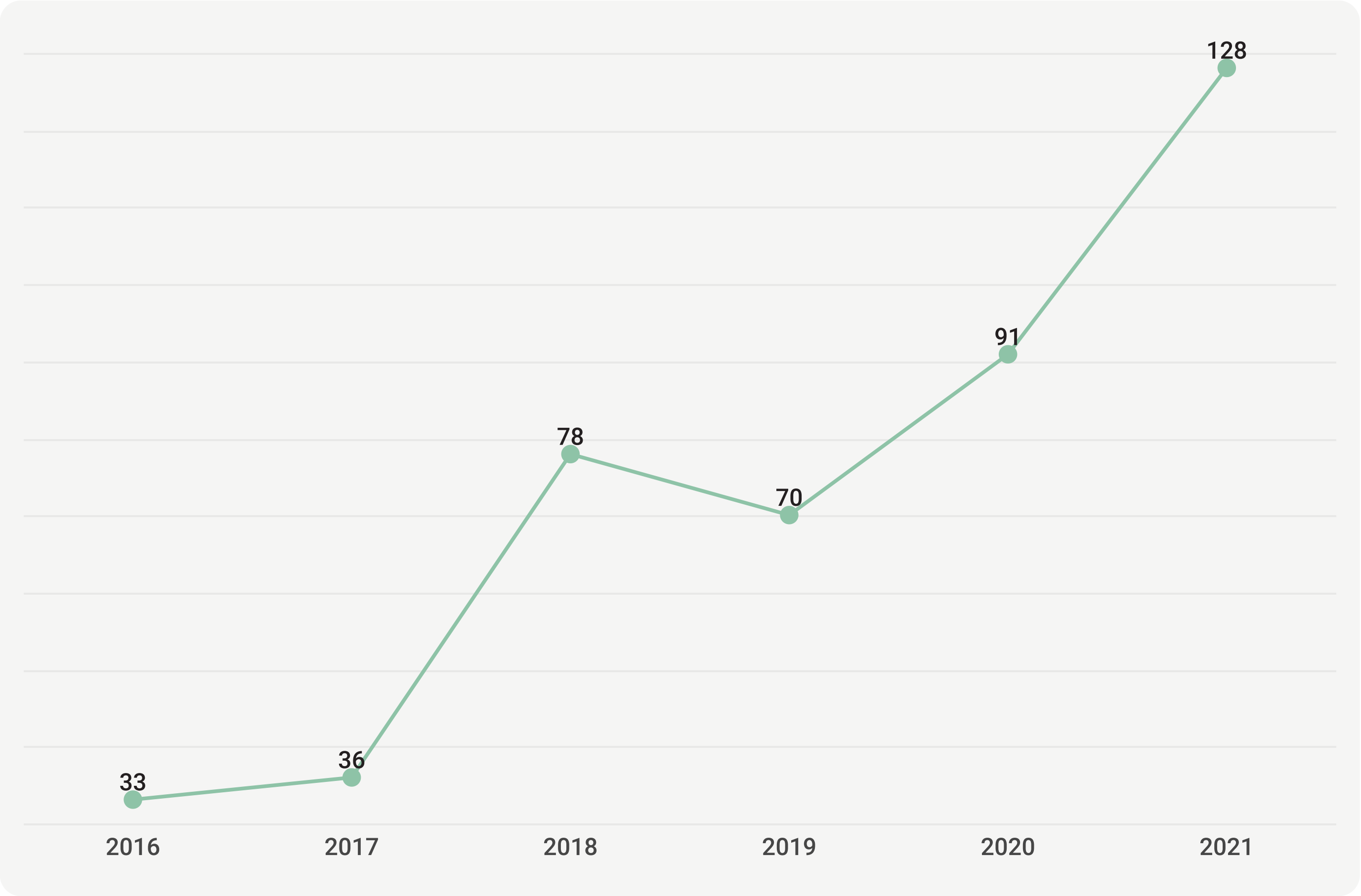


The number of new CWE-20: Improper Input Validation vulnerabilities discovered fell in 2019 and 2020 but increased slightly in 2021. Most ecosystems (JavaScript, Python, PHP) saw an increase in the number of new CWE-20 vulnerabilities in the past year. Java is the exception — the number of CWE-20 vulnerabilities has decreased every year since 2018.

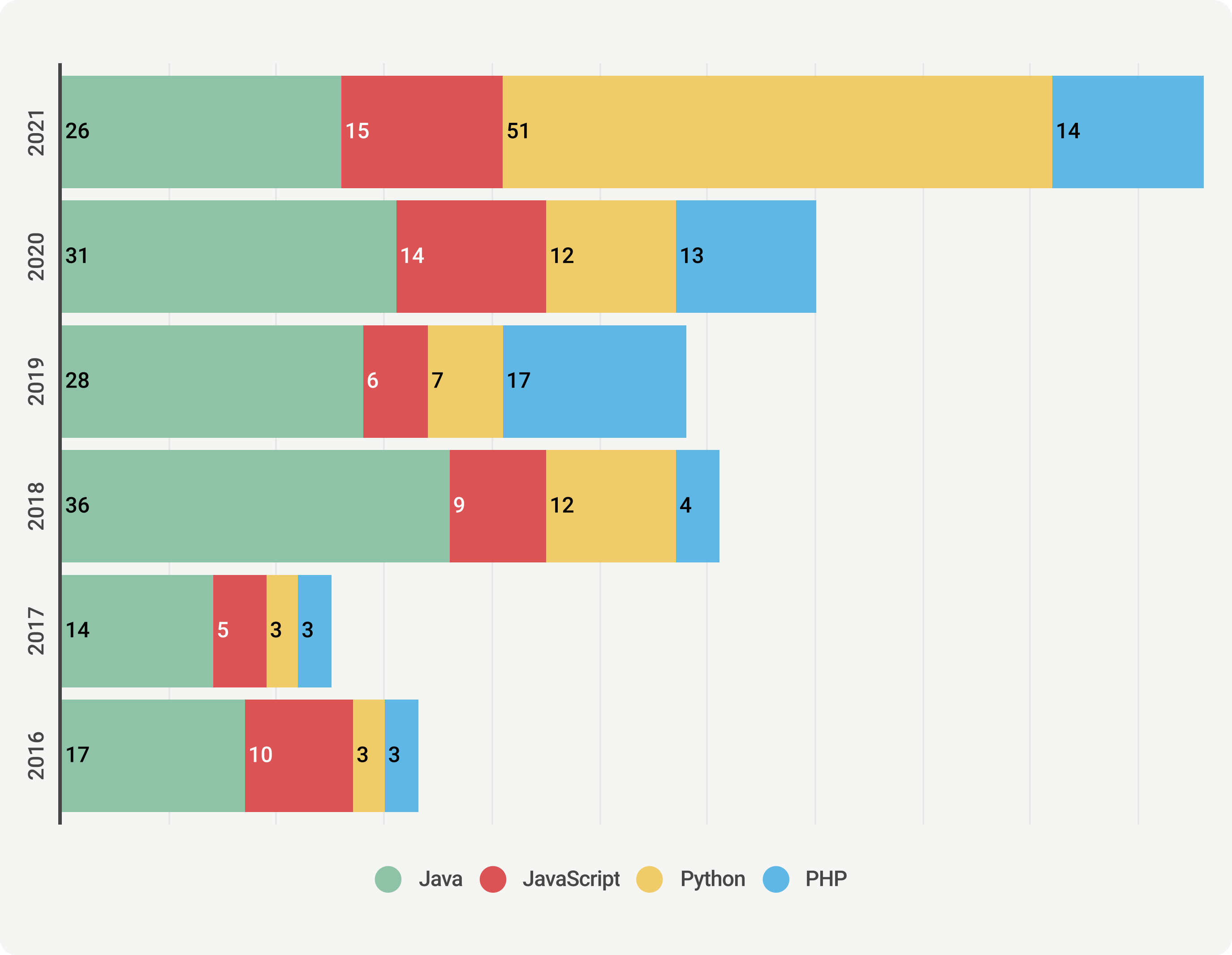
CWE-200: Information Exposure

> ***CWE-200: Exposure of Sensitive Information to an Unauthorized Actor***
>
> *The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.*
>
> *From MITRE*

CWE-200: Vulnerabilities Discovered by Year



CWE-200: Vulnerabilities by Ecosystem



There’s been a steady increase in the number of discovered CWE-200: Information Exposure vulnerabilities since 2017, with a 40% spike in 2021. Almost all of that increase can be attributed to the Python ecosystem.

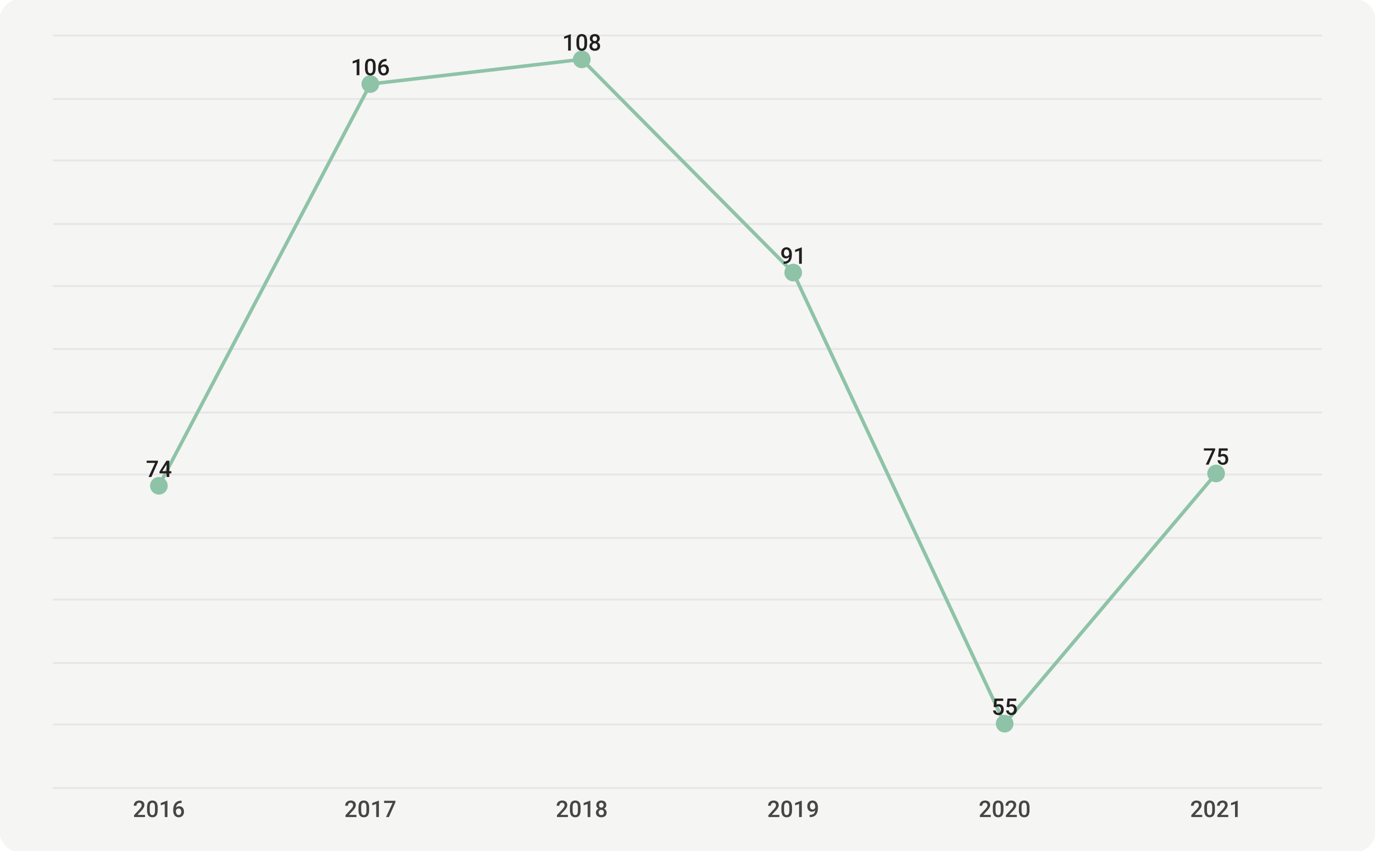
CWE-119: Buffer Overflows

CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer

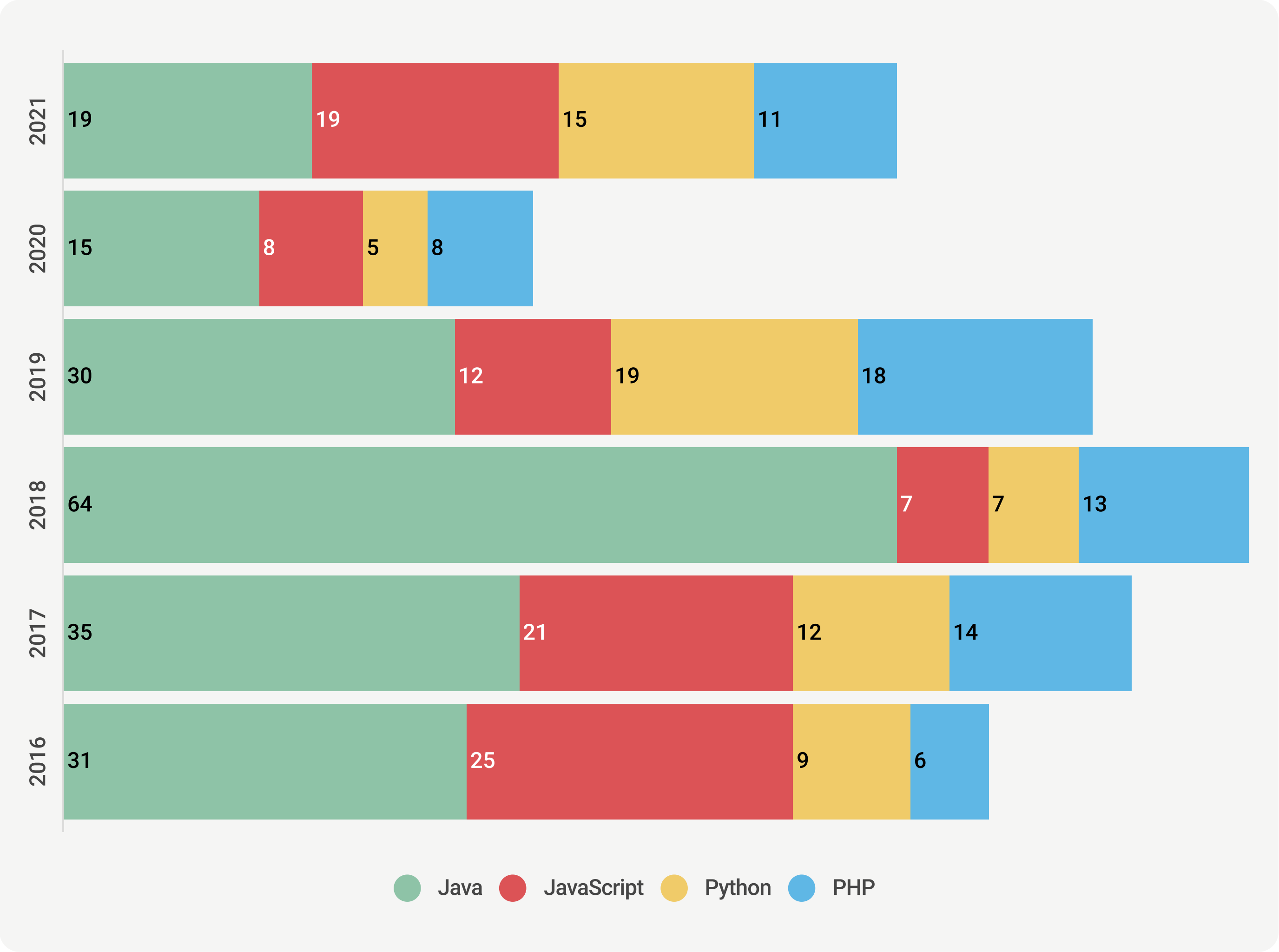
The software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer.

From MITRE

CWE-119: Vulnerabilities Discovered by Year



CWE-119: Vulnerabilities by Ecosystem

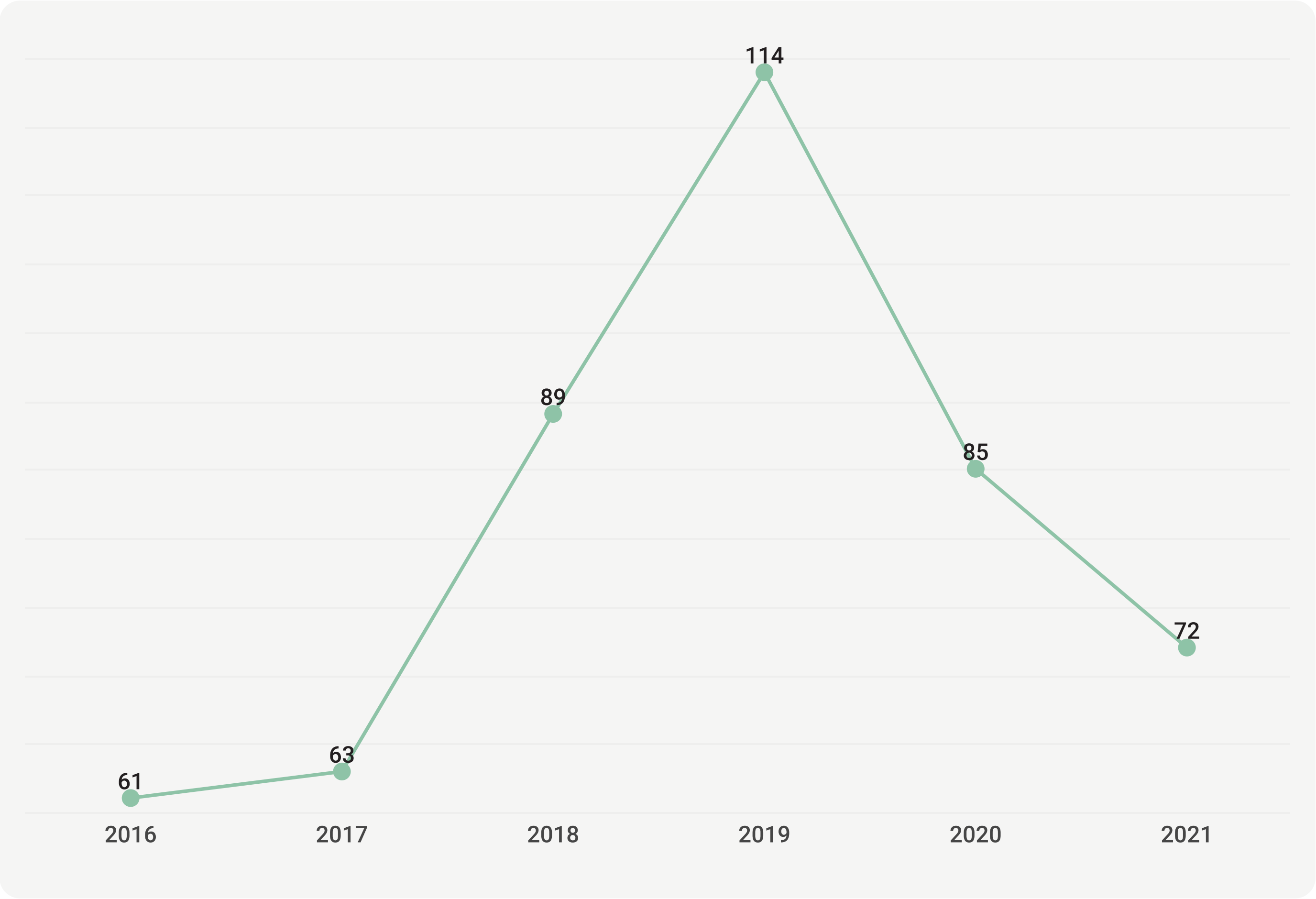


After peaking in 2018, the number of CWE-119: Buffer Error vulnerabilities fell in 2019 and 2020 before increasing again in 2021. Java accounted for a majority of the CWE-119 vulnerabilities, and its ecosystem trendlines mirror the overall trends for this vulnerability.

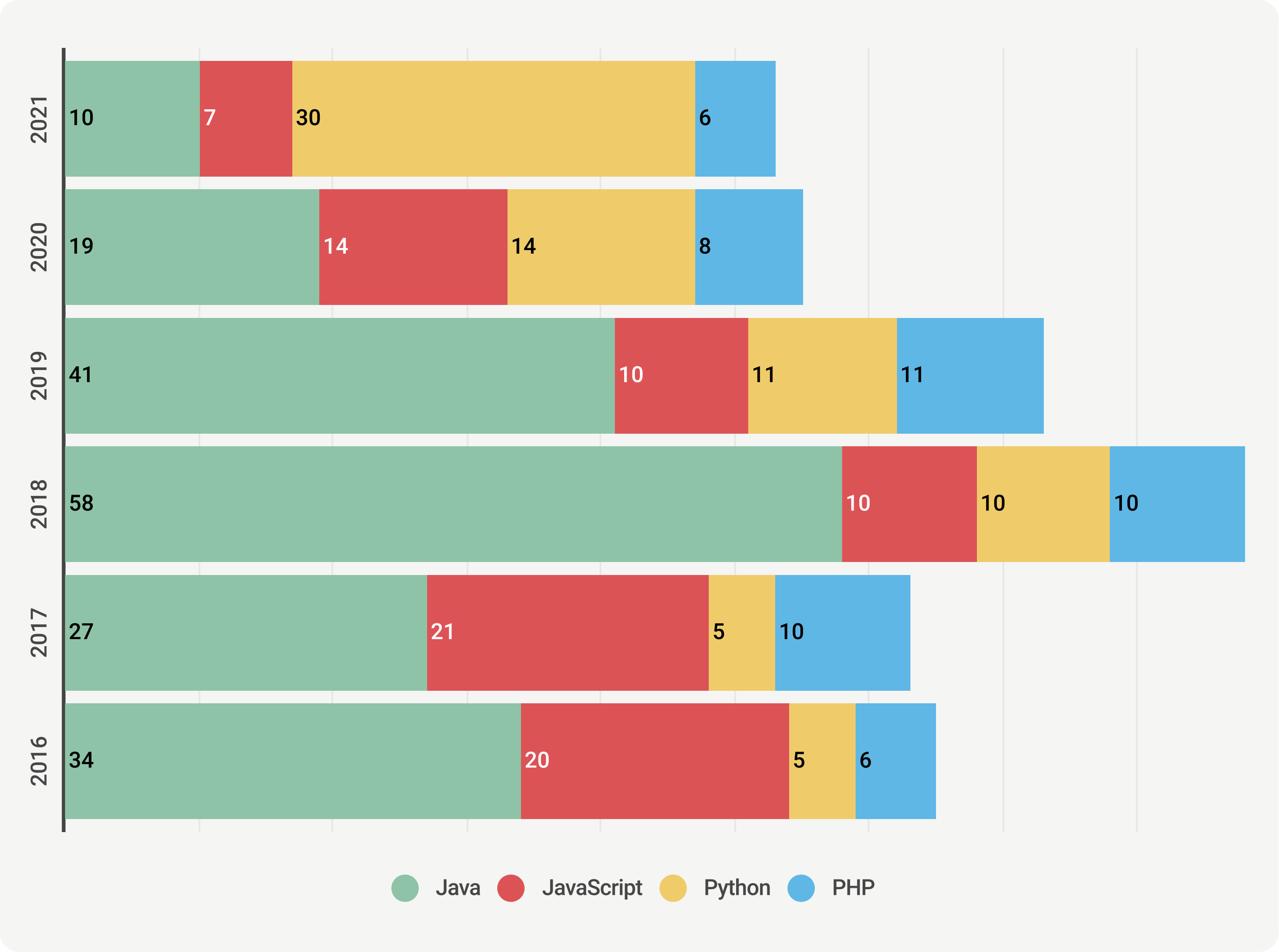
CWE-125: Out-of-Bounds Read

> ***CWE-125: Out-of-Bounds Read***
>
> *The software reads data past the end, or before the beginning, of the intended buffer.*
>
> *From MITRE*

CWE-125: Vulnerabilities Discovered by Year



CWE-125: Vulnerabilities by Ecosystem



After peaking in 2018, CWE-125: Out-of-Bounds Read has seen a modest downward trend in the number of new discovered vulnerabilities. Java used to be the ecosystem with the most discovered CWE-125 vulnerabilities, but Python overtook it in 2021.

Top 5 Vulnerabilities by Ecosystem

JavaScript	Java	Go	Ruby
CWE-79	CWE-79	CWE-79	CWE-20
CWE-119	CWE-20	CWE-125	CWE-125
CWE-20	CWE-119	CWE-119	CWE-79
CWE-125	CWE-125	CWE-200	CWE-119
CWE-200	CWE-200	CWE-20	CWE-787

PHP	Python	Rust	.NET
CWE-79	CWE-79	CWE-79	CWE-79
CWE-20	CWE-119	CWE-200	CWE-20
CWE-119	CWE-125	CWE-20	CWE-119
CWE-125	CWE-200	CWE-416	CWE-787
CWE-787	CWE-20	CWE-476 / CWE-119	CWE-125

It’s no surprise that CWE-79: Cross-Site Scripting errors were the most common across most ecosystems. This was the case for both new vulnerabilities discovered in 2021 and the number of total active vulnerabilities.

However, it’s important to note that the bulk of the CWE-79 errors were discovered a couple of years ago; Compared to previous years we’ve seen new instances of CWE-79 decrease across all ecosystems except Python.

CWE-199: Information Management Errors, CWE-20: Improper Input Validation, and CWE-125: Out-of-Bounds Read were common across all languages as well. CWE-

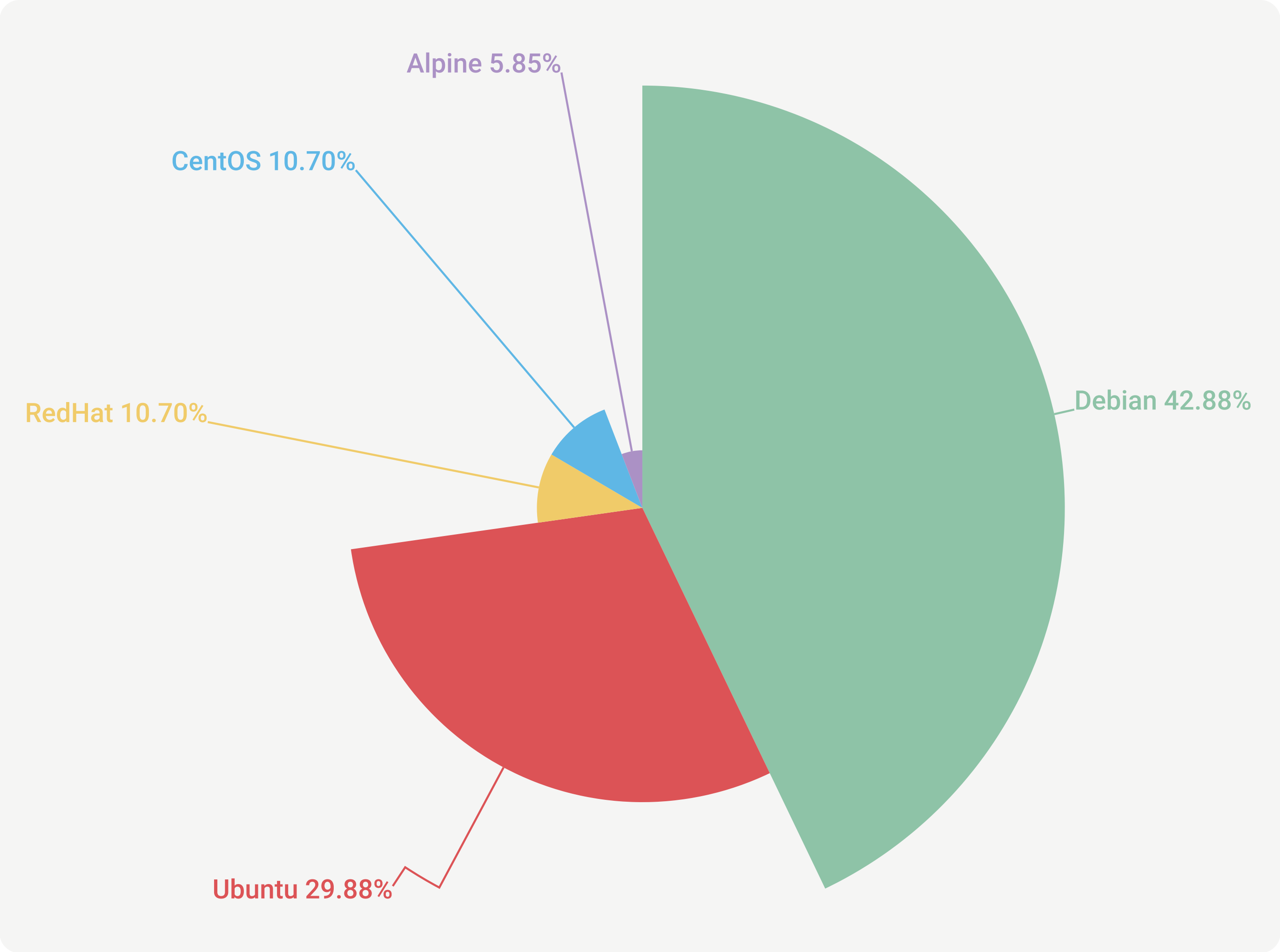
416: Use After Free and CWE-476: Null Pointer Dereference cracked the top-five most common vulnerabilities found in Rust, but they weren’t discovered in high numbers in other languages. This makes sense because Rust is a lower-level language that gives more user control over memory management.

Containers and Security

The rise of multi-cloud and hybrid-cloud environments has increased container adoption in many enterprises. With containers now widely used in network infrastructure and enterprise applications, container security has become a vital initiative.

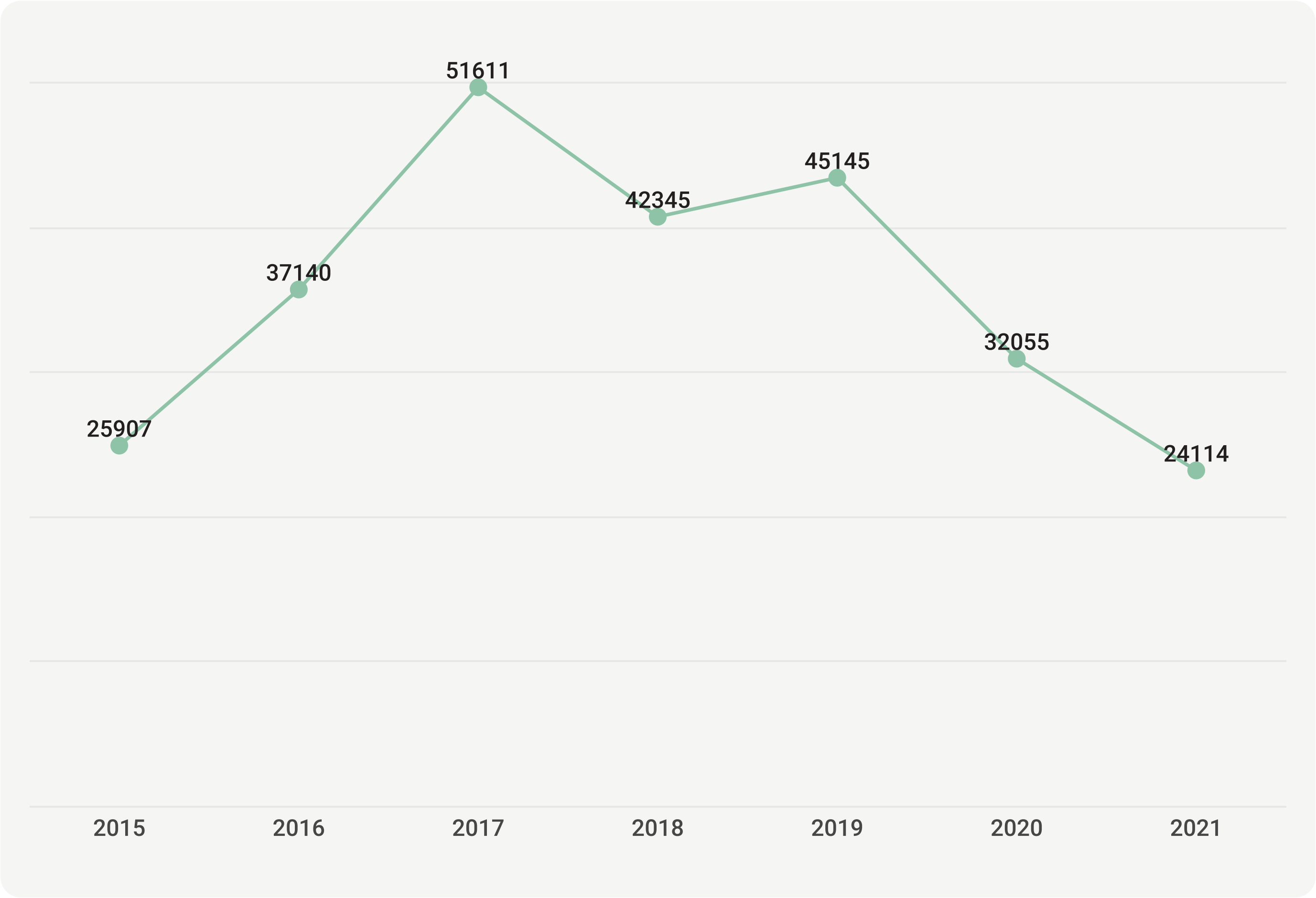
Containers are composed of predefined base layers and additional language-specific application layers that are added by the user. Linux distributions are among the most popular container base images, and this report will focus on vulnerabilities in the Linux ecosystem.

Vulnerabilities Across Linux Distributions



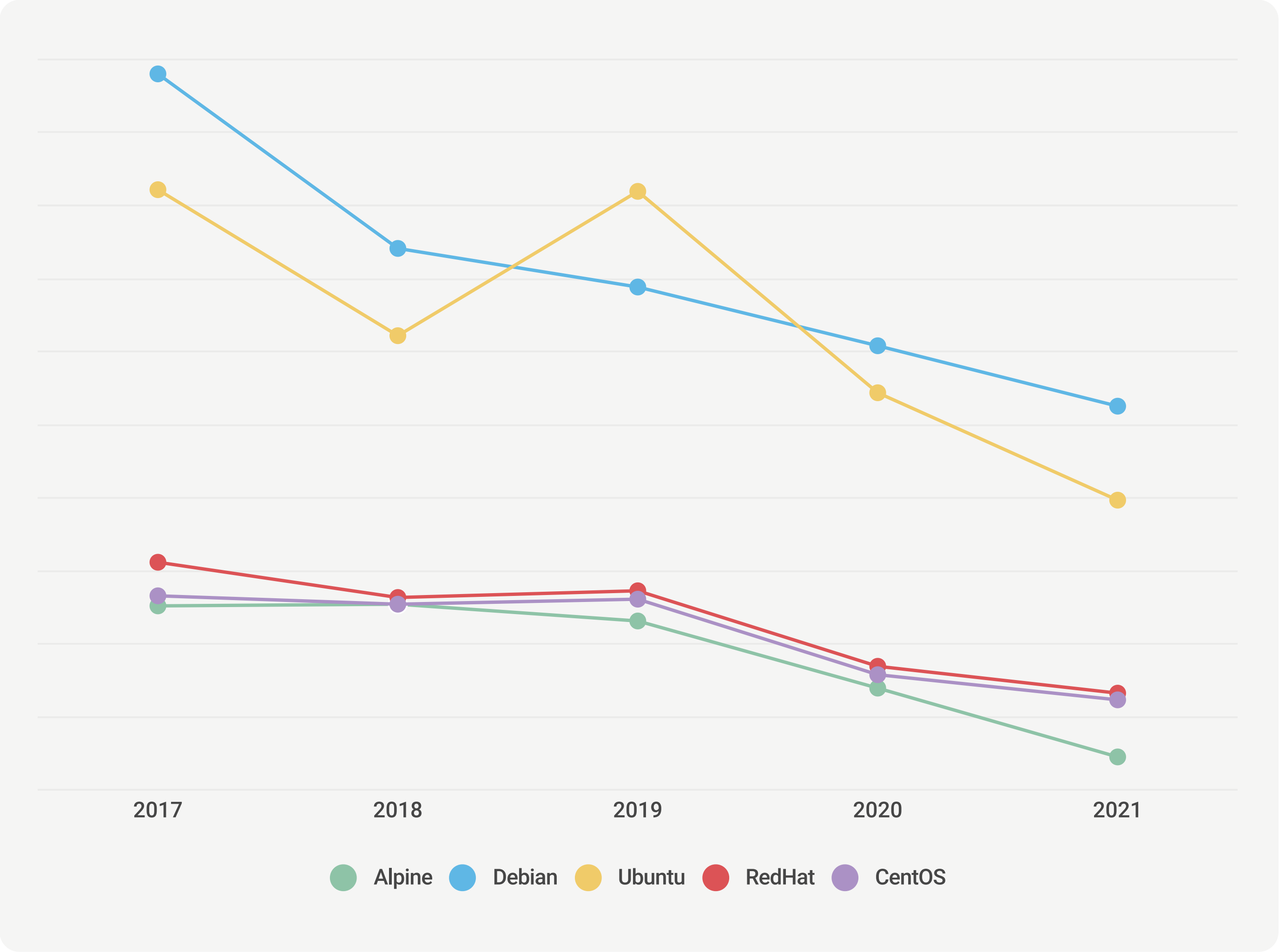
Debian, the most popular Linux distribution, also had the most number of total active vulnerabilities (and it accounted for a higher percentage of vulnerabilities than in 2020). Ubuntu is based on Debian, so it should also come as no shock that the two distributions have similar vulnerability counts. CentOS is based on Red Hat and both of these distributions have a similar number of vulnerabilities as well.Alpine, with its smaller footprint and built-in security protections, predictably had the lowest number of vulnerabilities.

Yearly Distribution of Vulnerabilities - Linux



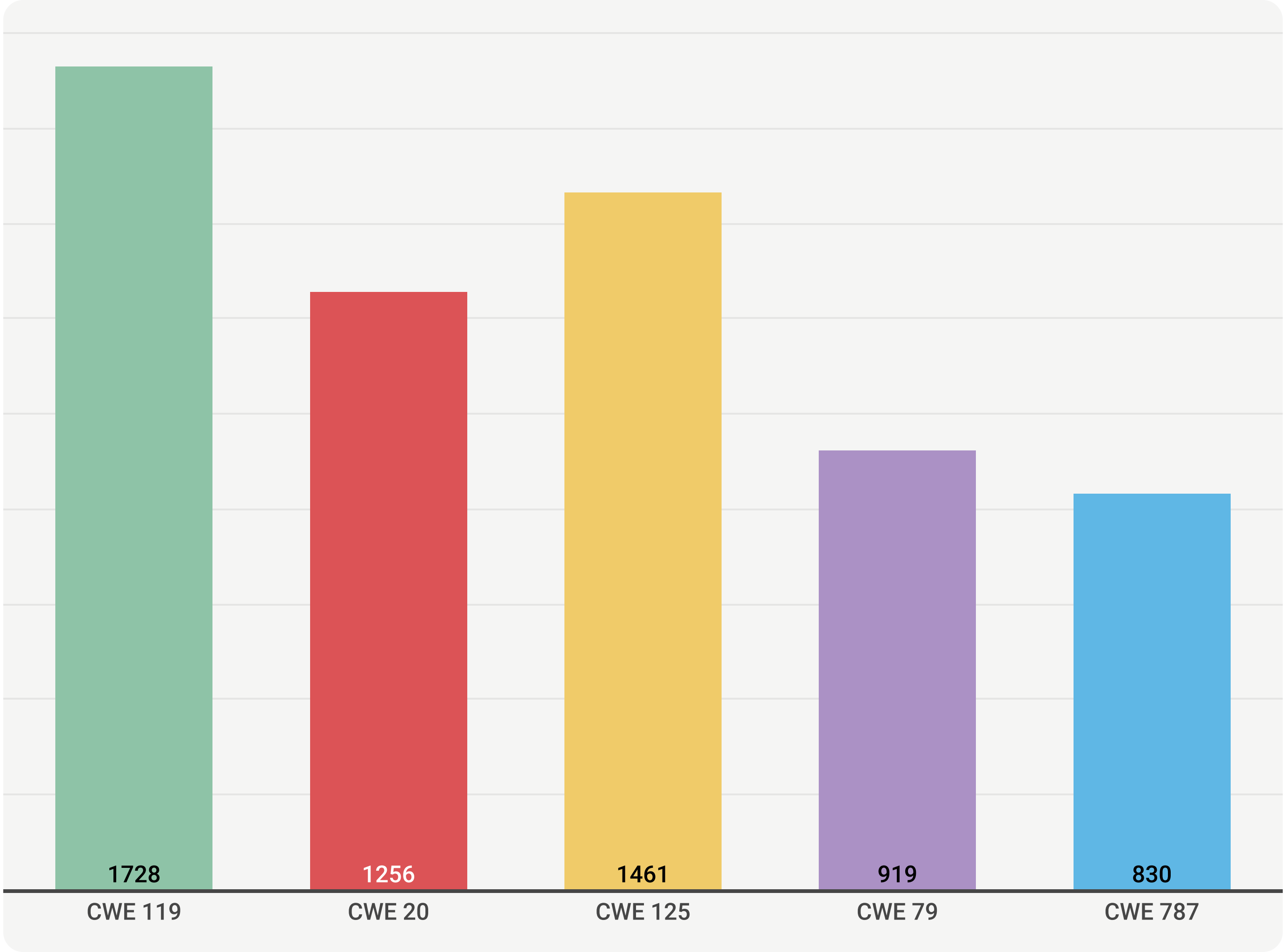
The number of vulnerabilities across the examined Linux distributions peaked in 2017. In the years since, there has been a downward trend in the number of vulnerabilities discovered. However, while the overall counts might be trending down, it is still essential that developers and security teams be diligent in scanning their Linux distributions to understand the composition and risk factor.

Distribution of Vulnerabilities Across Linux Ecosystems

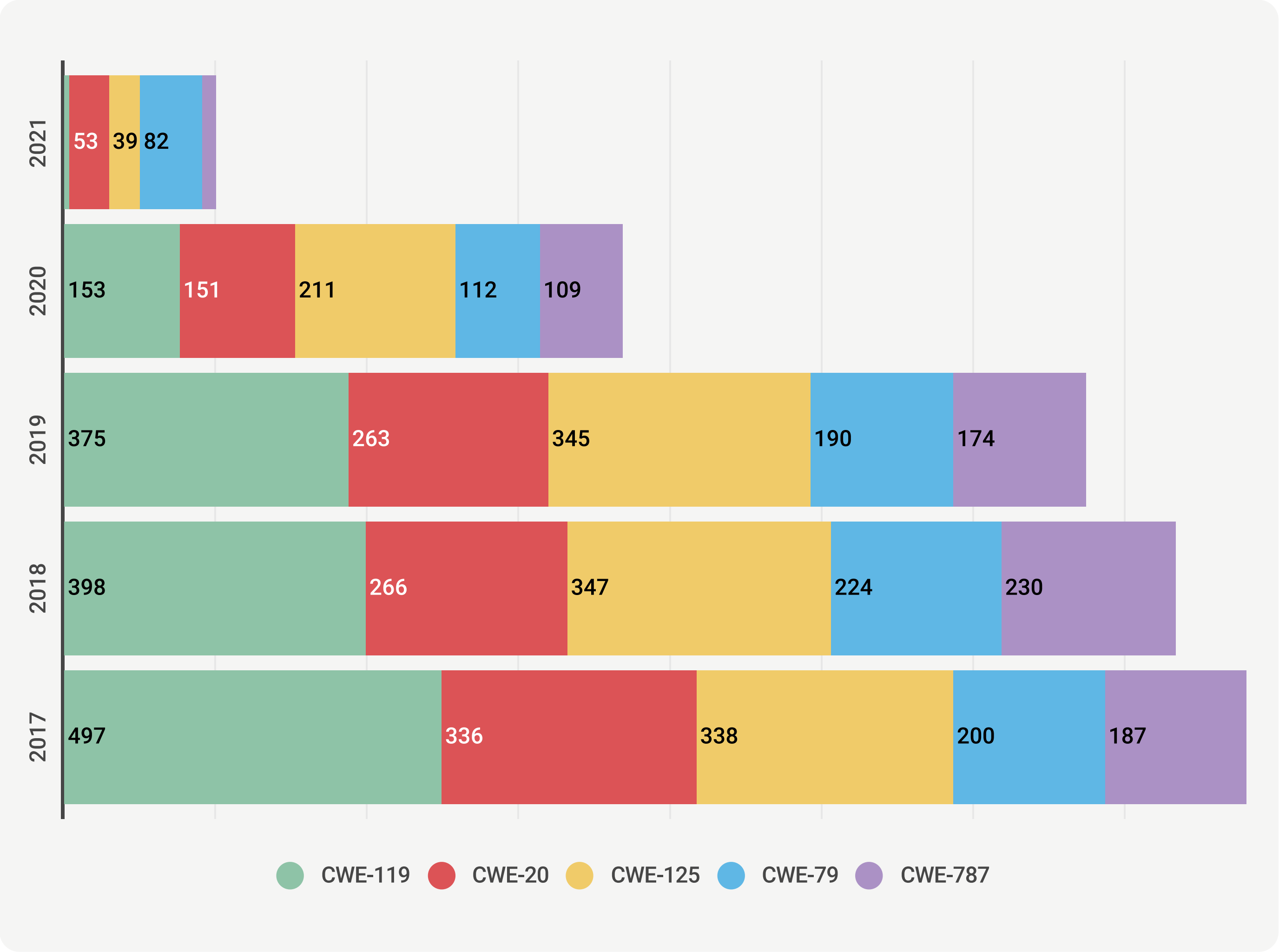


The numbers for the individual distributions reflected the overall trends: a drop in the number of new vulnerabilities discovered since 2017. However, it remains to be seen if this is a permanent trend borne out of successful security efforts. Alpine saw the biggest drop in the number of new vulnerabilities discovered among the examined distributions.

Most Common Vulnerabilities: Alpine

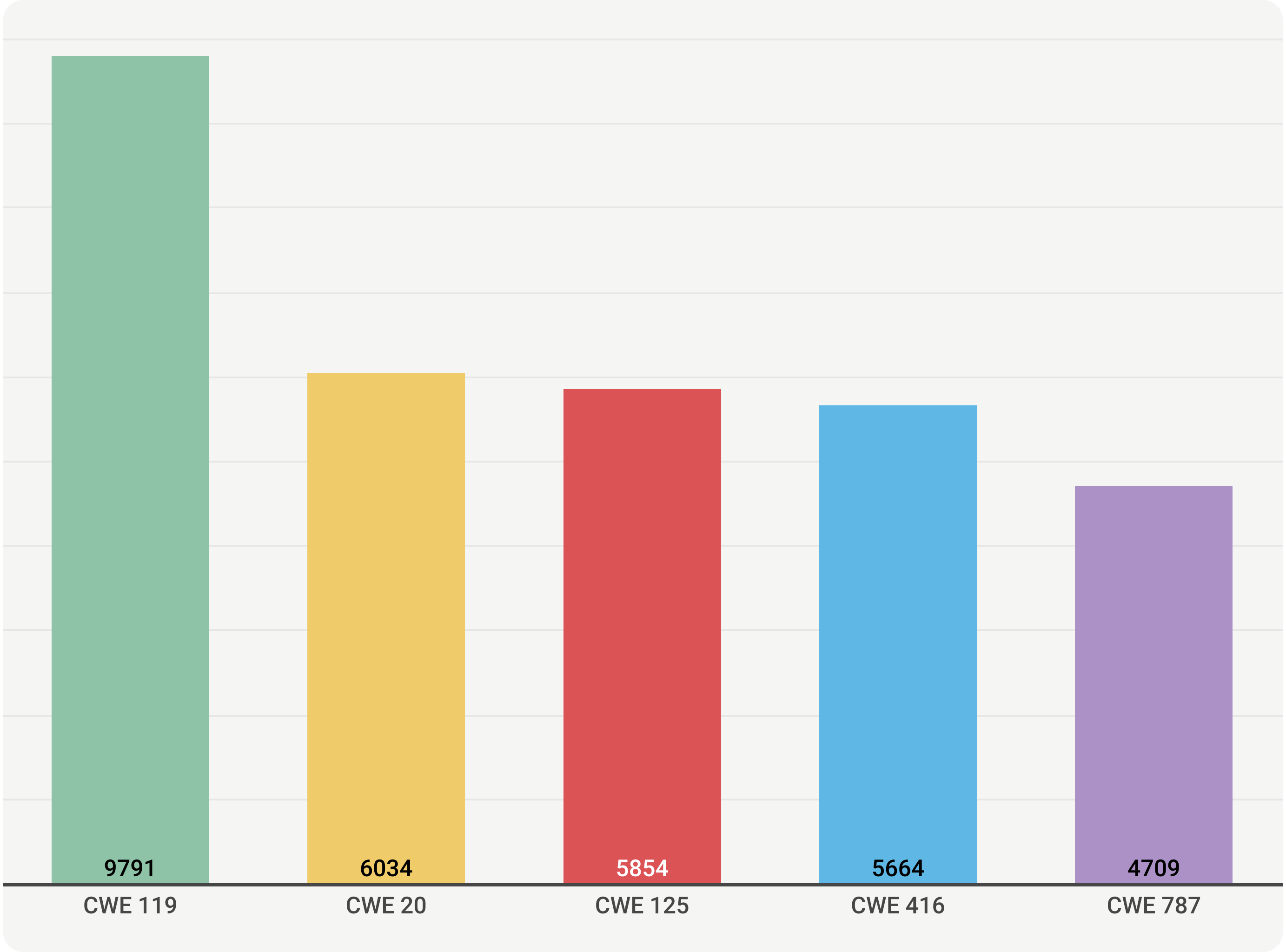


Yearly Vulnerability Trends: Alpine

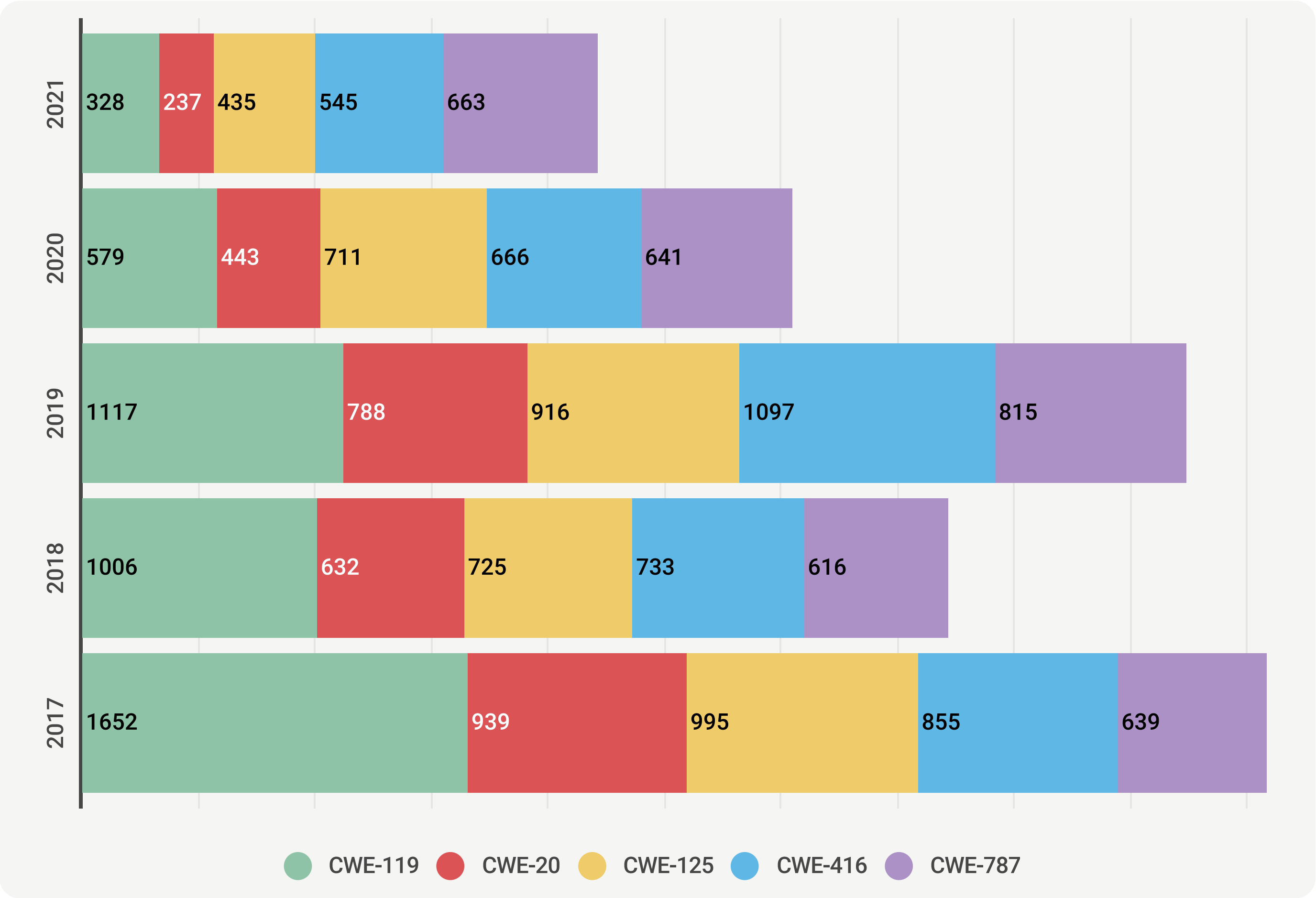


While CWE-119: Buffer Overflows and CWE-125: Out-of-Bounds Read are overall the most common vulnerabilities in Alpine, we can see that there were very few new instances of CWE-119 or CWE-125 discovered in Alpine in 2021. In the past year, Alpine had more new CWE-79: Cross-Site Scripting and CWE-20: Improper Input Validation vulnerabilities than any other category.

Most Common Vulnerabilities: Ubuntu



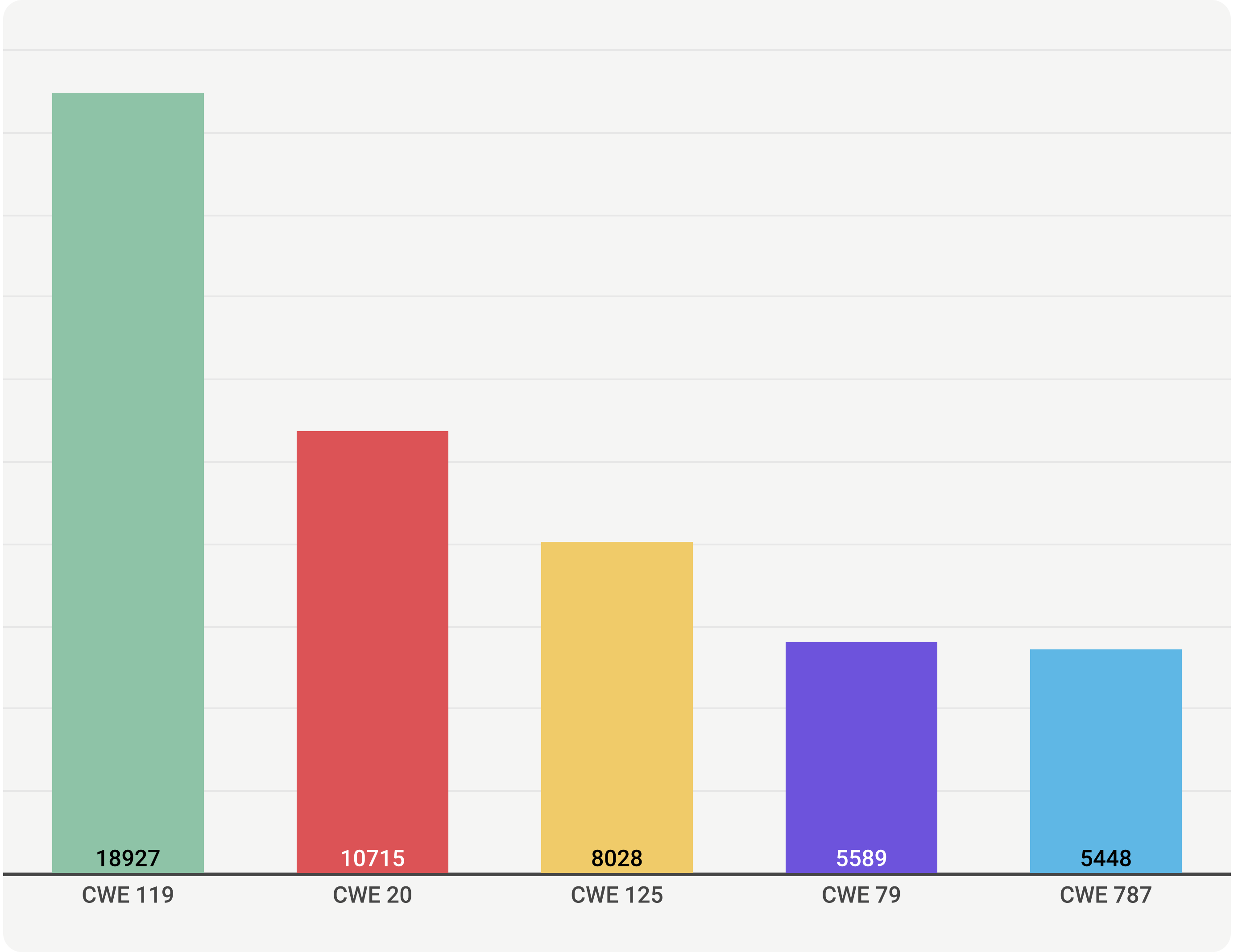
Yearly Vulnerability Trends: Ubuntu



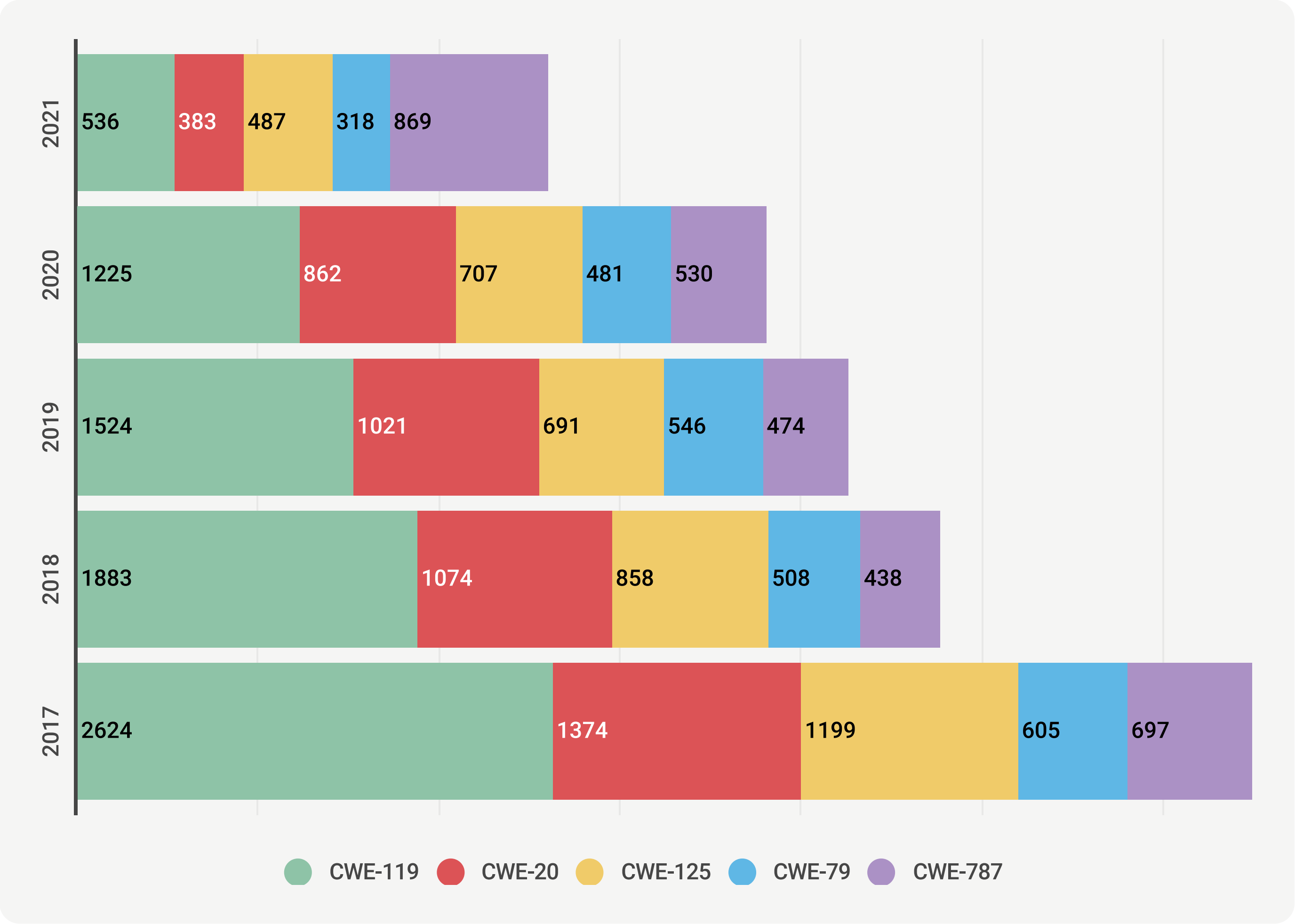
Historically, CWE-119: Buffer Overflows has been the most common vulnerability in Ubuntu, but the number of new CWE-119 issues dropped sharply in 2020 and 2021. In contrast, there was a small uptick in new CWE-787: Out-of-Bounds Write vulnerabilities in 2021.

Despite being quite prevalent in previous years, CWE-119: Buffer Overflows and CWE-20: Improper Input Validation were the least commonly found new vulnerabilities in 2021. Over the past couple of years, we have seen more new CWE-125: Out-of-Bounds Read, CWE-787: Out-of-Bounds Write, and CWE-416: Use After Free instances than any other CWE.

Most Common Vulnerabilities: Debian



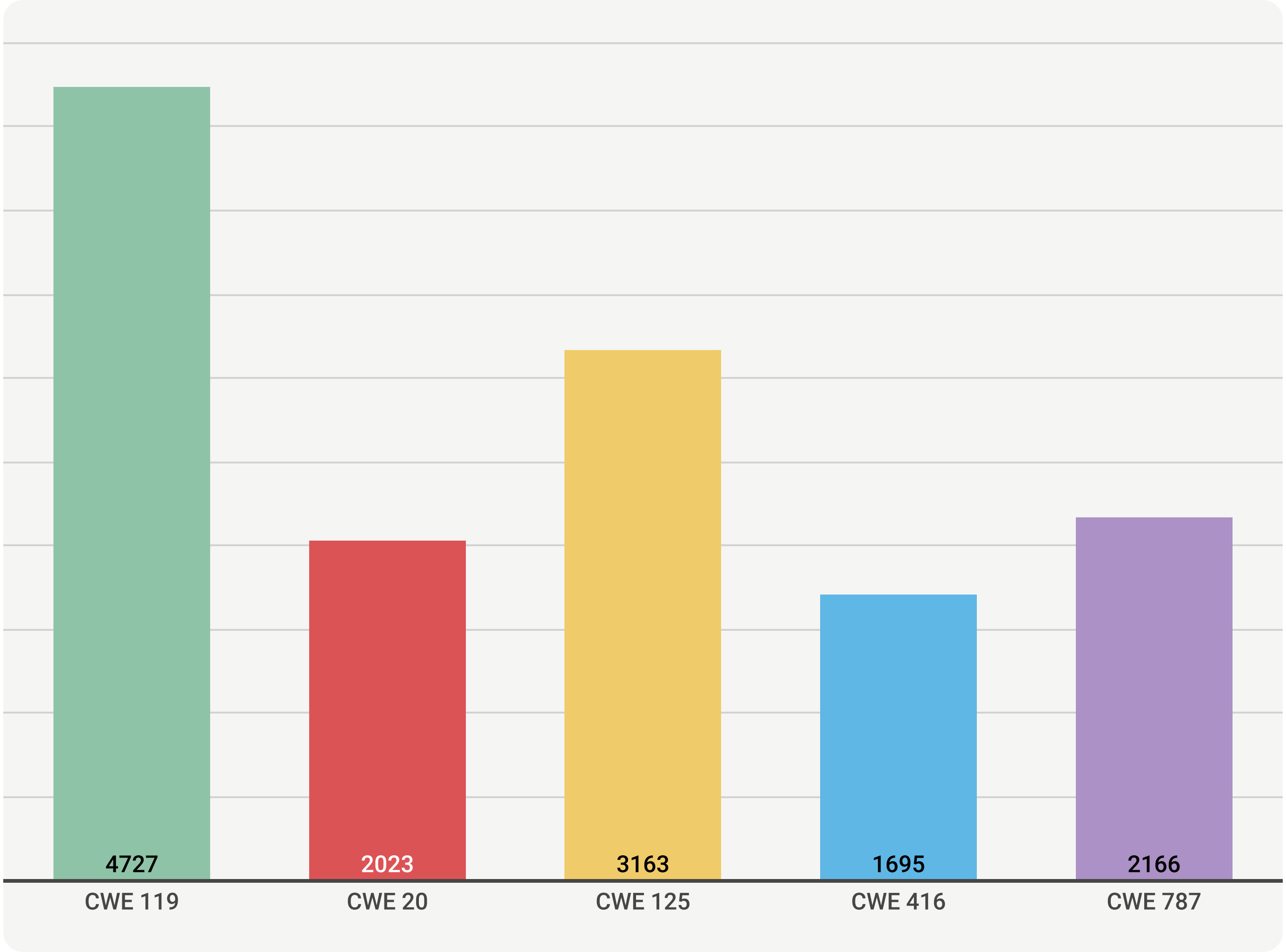
Yearly Vulnerability Trends: Debian



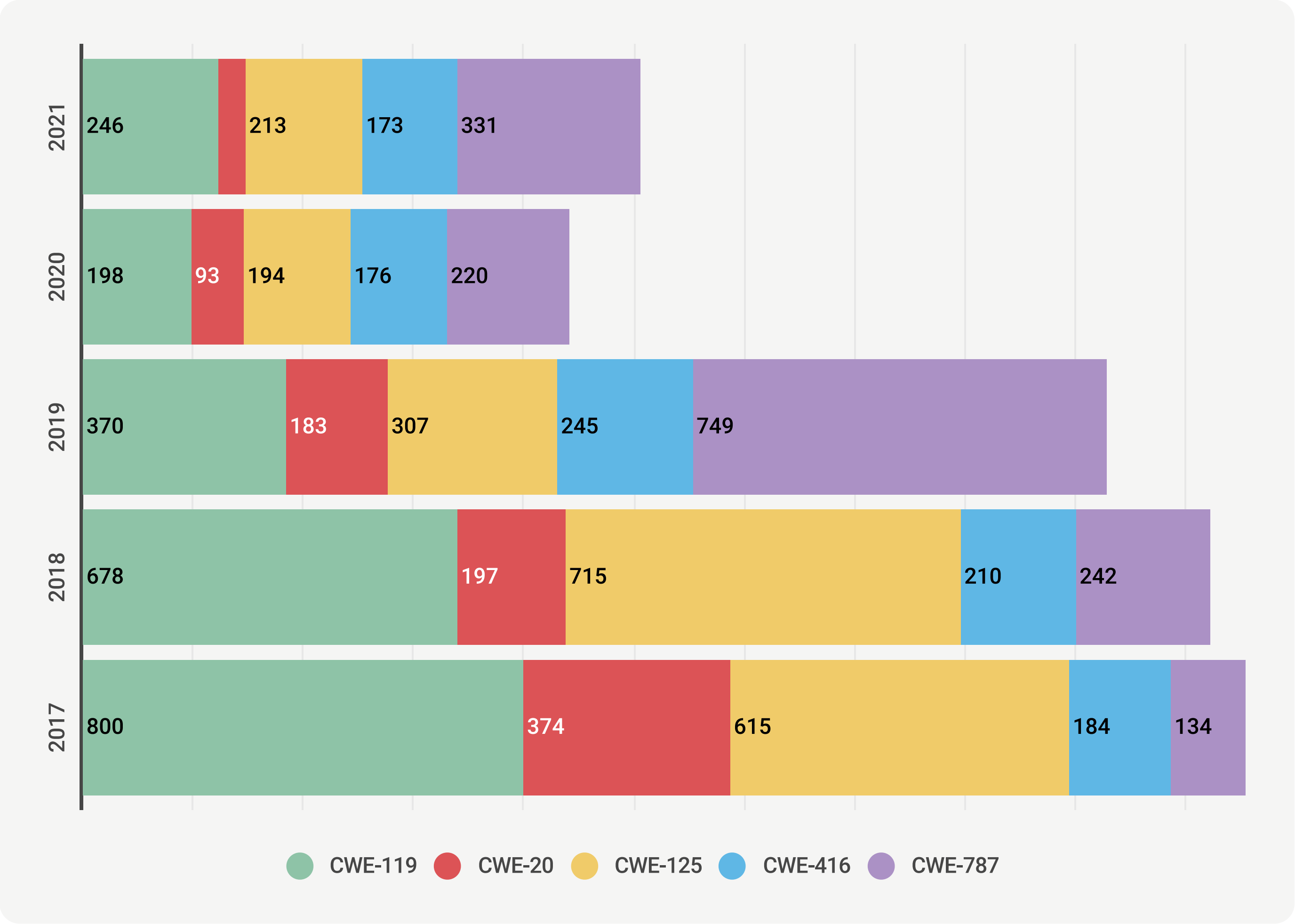
In 2021, Debian saw a steep drop in the number of new CWE-119: Buffer Overflows vulnerabilities discovered — prior to 2021, CWE-119 was overwhelmingly the most common category of vulnerability in Debian. The distribution also shared similar movements with CWE-119: Improper Input Validation and CWE-20: Buffer Overflows, which were historically found in large numbers but declined significantly in 2021.

Finally, in contrast to the other vulnerabilities analyzed, CWE-787: Out-of-Bounds Write saw a considerable increase in new vulnerabilities in 2021.

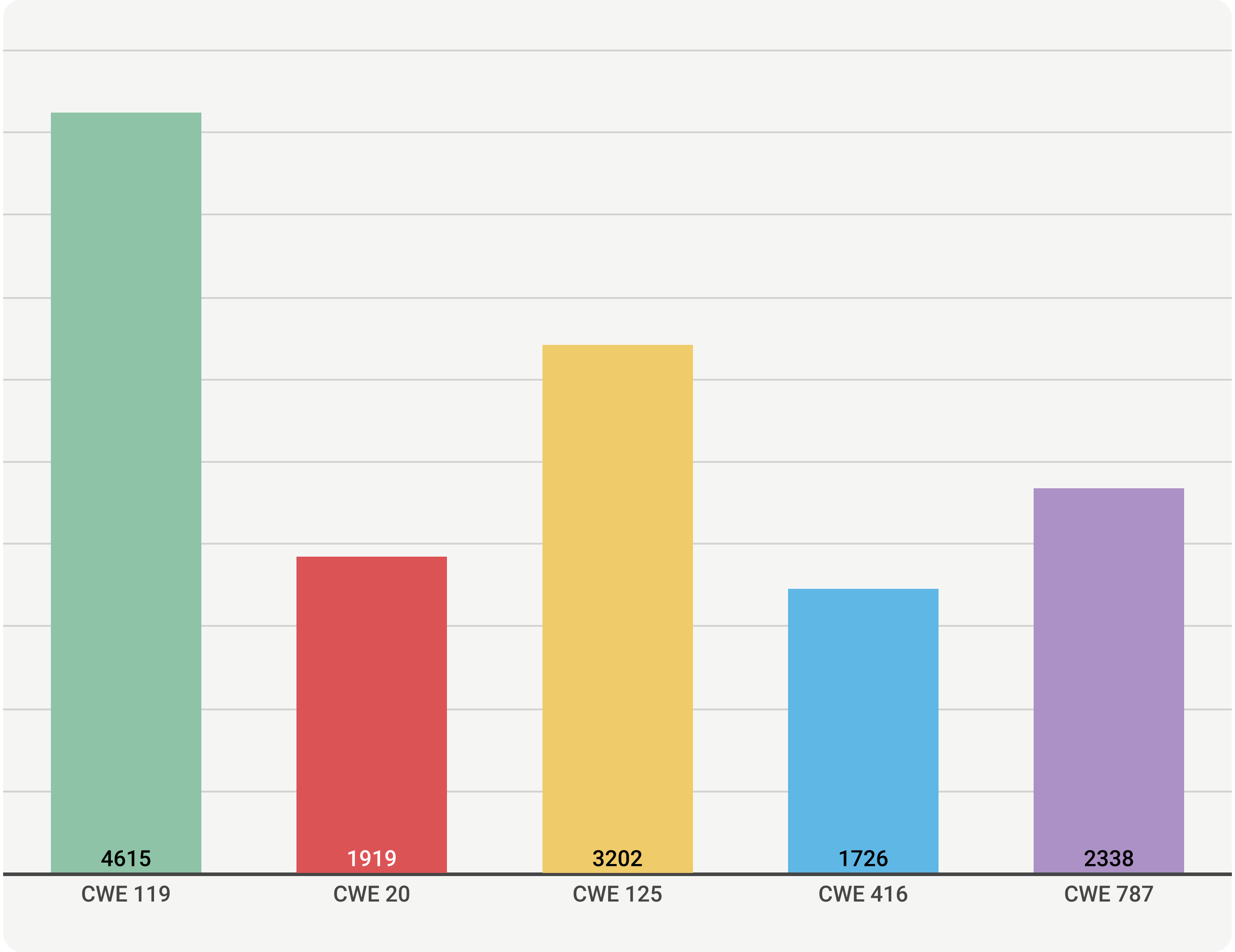
Most Common Vulnerabilities: RedHat



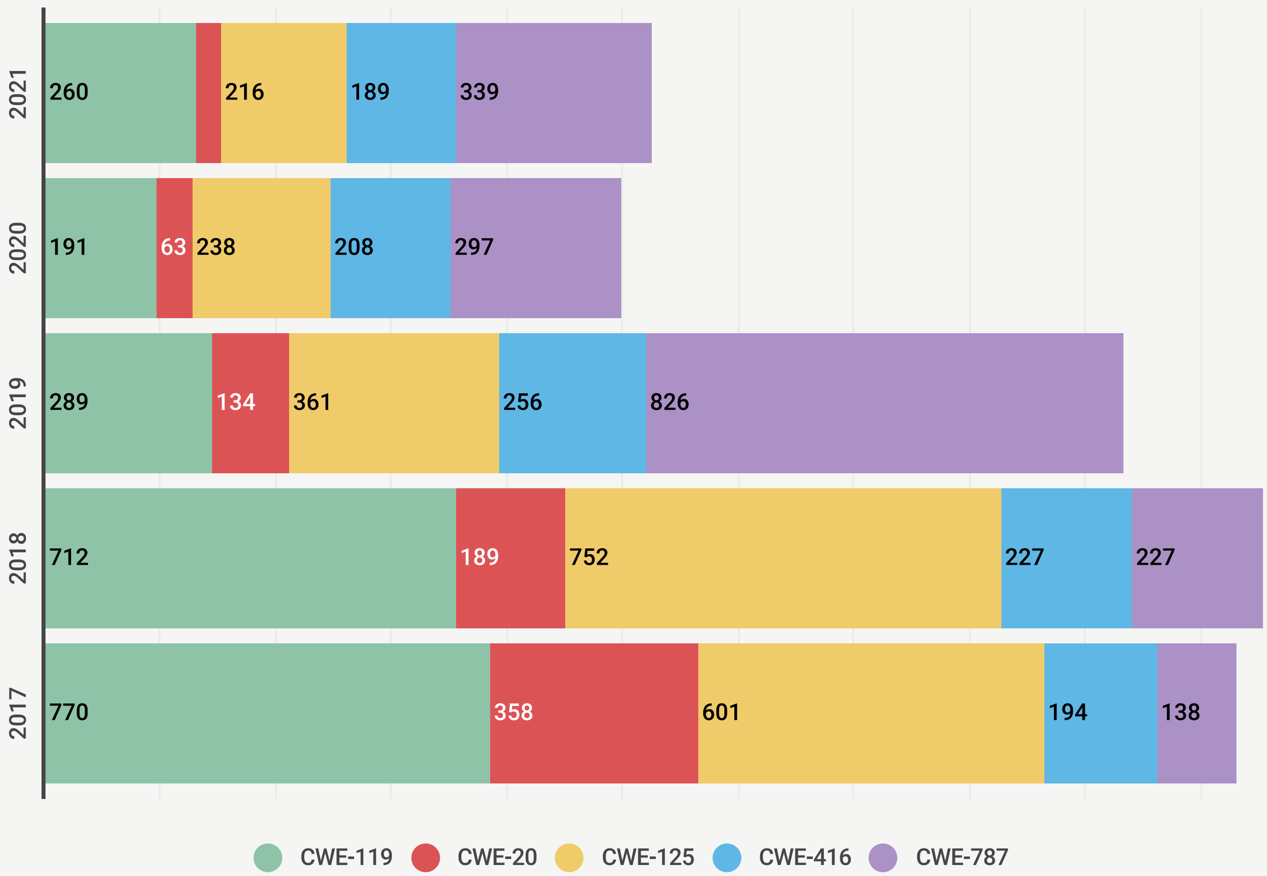
Yearly Vulnerability Trends: RedHat



Most Common Vulnerabilities: CentOS



Yearly Vulnerability Trends: CentOS



RedHat and CentOS experienced similar trends in the year-over-year popularity of common vulnerabilities, such as historically common vulnerabilities CWE-119: Buffer Overflows and CWE-20: Improper Input Validation giving way to CWE-787: Out-of-Bounds Write errors.

The increase in new CWE-787: Out-of-Bounds Write errors across all distributions suggests developers should use defensive code to guard against buffer overflows, along with static and dynamic analysis tools to detect them.

Conclusion

Although 2021 will be remembered for several serious security incidents, especially the series of Log4J vulnerabilities, the news wasn't all bad. New vulnerabilities actually decreased by an average of 5% across most popular ecosystems. Python, which saw more than a two-fold increase in new vulnerabilities, was the exception.

Similarly, the total count of new vulnerabilities across Linux distributions decreased by 32%, with Alpine registering the sharpest decrease in new vulnerabilities discovered.

Of course, despite this reduction, application security remains a major concern. And while there's no single, foolproof solution to secure software development, organizations would benefit from a proactive approach to identifying and remediating vulnerabilities. This includes:

- ✓ Security training that educates developers on common attack vectors for vulnerabilities
- ✓ Maintaining an in-depth and accurate inventory of their open source
- ✓ Developer and security teams working in tandem to identify and mitigate vulnerabilities as early as possible
- ✓ Scanning code for vulnerabilities continuously in the software development lifecycle

You can also visit our website for more information on open source vulnerability management.

About FOSSA

Up to 90% of any piece of software is from open source, creating countless dependencies and areas of risk to manage. FOSSA is the most reliable automated policy engine for security management, license compliance, and code quality across the open source stack. With FOSSA, engineering, security, and legal teams all get complete and continuous risk mitigation for the entire software supply chain, integrated into each of their existing workflows.

FOSSA enables organizations like Uber, Zendesk, Twitter, Verizon, Fitbit, and UiPath to manage their open source at scale and drive continuous innovation. Learn more at https://fossa.com.