



Red Hat Product Security risk report

2020

Contents

Introduction	1
14+ years of reporting open source risk	2
Sources and methodologies	5
Vulnerabilities	6
Vulnerability trends	12
Which issues were branded and which really mattered in 2020	20
The open source ecosystem	24
Conclusion	26

2020 at a glance:

3,011 security issues were reported to Red Hat Product Security (up from 2019).

2,040 CVEs were addressed throughout 2020, a 55% increase from 2019.

1,523 Red Hat Security Advisories were issued, a record increase over previous years.

53 Critical Advisories addressed 19 Critical vulnerabilities.

31% of Critical issues were addressed within one business day—slightly less than last year.

89% of Critical issues were addressed within one week—slightly ahead of last year.

100% of Critical issues were addressed within 31 days of public release—exceeding 2019's performance.

11% of Important issues were addressed within one business day—slightly down from last year.

28% of Important issues were addressed within one week—roughly on pace with 2019.

64% of Important issues were addressed within 31 days—exceeding 2019's delivery.

Introduction

The 2020 and 10th edition of the Red Hat® Product Security risk report is an overview of security vulnerabilities that impacted Red Hat products for the 2020 calendar year. In this report, security vulnerabilities publicly announced throughout the last calendar year and the data and metrics produced for these vulnerabilities across all of our portfolios are reviewed. High-impact, high-profile events that affected Red Hat offerings and deserve more attention than many of the others are also reviewed.

In this report, *product* means a Red Hat offering listed at <https://access.redhat.com/products> and the associated version(s) available in 2020. All security related issues that impact one of those products are documented and assigned a [Common Vulnerabilities and Exposures \(CVE\)](#) identifier and a Red Hat [severity](#) rating by our [Red Hat Product Security](#) team. Red Hat Product Security includes Red Hat's [Product Security and Incident Response Team \(PSIRT\)](#) that has been serving Red Hat, our subscribers, communities and partners since [September 2001](#).

If we fix a general bug that later turns out to have a security implication, we retroactively assign a CVE name to that issue. Every fixed CVE has an entry in our public database in the [Red Hat Customer Portal](#) and a public bug report with more technical detail. In this report, we will use *vulnerabilities* and *CVEs* interchangeably. We report issues that represent a meaningful risk to users of our offerings and describe exploitable issues. Data used to create this report is available from public data collected and analyzed by Red Hat Product Security.

Every vulnerability reported to Red Hat Product Security is reviewed and analyzed by our team of open source security specialists. These engineers understand how our offerings are composed, curated, hardened, packaged, delivered, and used by our customers. This breadth of experience and insight into Red Hat product engineering's security-focused supply chain practices helps provide critical insights into the potential impact of these vulnerabilities on our products and services.

Red Hat has more than 19 years of focused experience through the dedicated Red Hat Product Security team. We have worked through the evolution of physical servers, virtual machines, virtual images in the cloud, and the decomposition of legacy applications into microservices and containers, moving processing out to the edge and beyond. Along this journey, we have forged deep bonds with the open source community, which has earned us wisdom and insights into the challenges open source faces with security.

14+ years of reporting open source risk

Red Hat began reporting on the vulnerabilities discovered within components of our portfolio in 2005. Mark Cox authored a [blog](#) and later the Red Hat Summit presentation, "[A Year of Red Hat Enterprise Linux® 4](#)," in the Spring of 2006. Back then, we recognized the need for providing Red Hat Enterprise Linux users with an idea of the potential severity and impact of a vulnerability. In his report, Mark also looked at the public exploits available, many of which were already being mitigated by security technologies built into RHEL such as stack protections and SELinux." In 2006, the Linux kernel had the most impactful vulnerabilities and still has the most impactful vulnerabilities to this day.

During this time, many [major media outlets](#) wrote articles asking "Which OS was better?". These articles portrayed closed-source vendors favorably in their remediation of Critical vulnerabilities over open source software. This media attention spurred Red Hat to talk publicly and frequently about the real data around open source vulnerabilities.

In June 2007, Red Hat released the next major update of the report. Mark reviewed CVE data points and trends from the first three months after the Red Hat Enterprise Linux 5 release, and he recapped the first two years of [Red Hat Enterprise Linux 4 risks](#).

Back then, we only had three Critical vulnerabilities to address in the release. Things were much simpler, with a handful of solutions to support and a relatively manageable volume of vulnerabilities to triage and analyze. The Red Hat Product Security team (then known as Red Hat Security Response team) only had seven people looking over about 25 products. While the team was very busy, addressing vulnerabilities was manageable with a small team of experts. Red Hat had been producing enterprise, open source software since 1993, but we only started publishing official security advisories in 1999.

When we released Red Hat Enterprise Linux 5.1 in November 2007, Mark revised the style by adding new piechart-free graphs and began creating the risk report to be what it is today. With each iteration, much like the open source code we support, the report became better and more useful, with interesting observations and insights. The message, “remaining vigilant and up-to-date,” was our charter for the report from the beginning. We realized that the open source ecosystem was constantly evolving and changing, and to keep our products resilient, we needed to track, document, and communicate.

In Spring 2008, we adopted the Risk Report title.

In the following years, not only did we write about the vulnerabilities discovered in our products, but we also included topics of concern in the industry. These topics included [discrepancies in third-party scanners](#) (a problem that still persists, which we have written about several times) and [differences in data](#) between the authoritative Red Hat Product Security scores and third-party vulnerability aggregators like [NVD](#).


The 2015 report transitioned from being a Red Hat Enterprise Linux-only conversation to a broader, more portfolio-view of the solutions we provide. Customers demanded more enterprise, open source solutions to power their business, so the Red Hat product portfolio evolved to include container solutions, automation and management technologies, and developer tools.

Risk report update: April to October 2015

🕒 Mark Cox published on November 4 2015 at 6:45 PM, last updated November 25 2015 at 3:41 PM

English ▾

In April 2015 we took a [look at a years worth of branded vulnerabilities](#), separating out those that mattered from those that didn't. Six months have passed so let's take this opportunity to update the report with the new vulnerabilities that mattered across all Red Hat products.



[ABRT](#) (April 2015) CVE-2015-3315:

ABRT (Automatic Bug Reporting Tool) is a tool to help users to detect defects in applications and to create a bug report. ABRT was vulnerable to multiple race condition and symbolic link flaws. A local attacker could use these flaws to potentially escalate their privileges on an affected system to root.

Figure 1. Risk report update: April to October 2015

While we had written risk reports at the time, the 2015 edition became the first “Red Hat Product Security Risk Report”

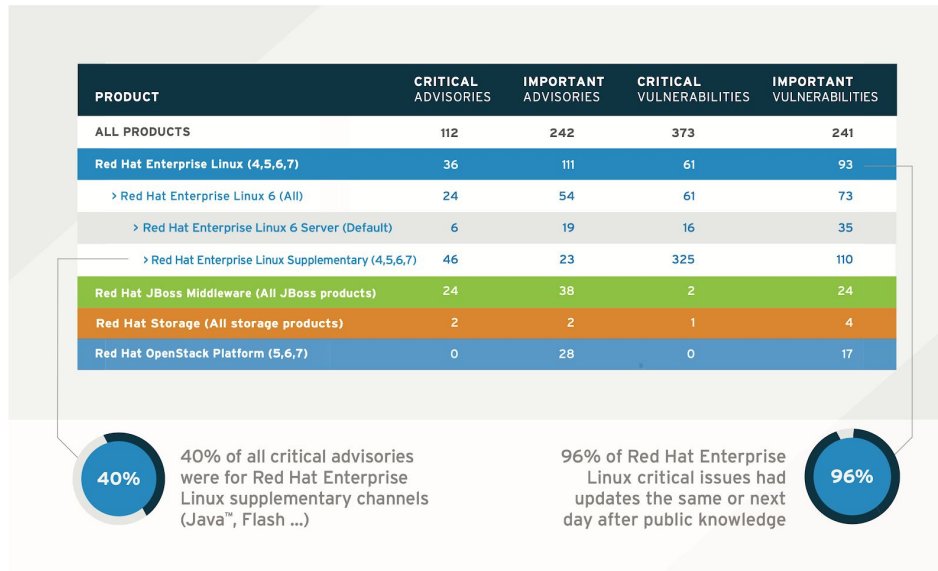


Figure 2. Red Hat security advisories and vulnerabilities 2015

With the larger portfolio view, we could upgrade to some infographics to compare and contrast the different areas of open source software we supplied. In 2015 we also began our [Customer Security Awareness \(CSAw\)](#), a special handling process Red Hat Product Security runs to help ensure all of our internal and external stakeholders have as much information about the flaws that generate media attention. CSAw helps us raise awareness about high-profile or critically severe vulnerabilities for our subscribers and the community.

In previous years, we noted the trend of the sensational naming of vulnerabilities that continues today. [CERT/CC](#) have taken this trend to extremes lately to [help highlight the silliness of a scary name](#). We started to talk more directly about the marketed and branded security flaws and included a section about which branded flaws had considerable impact to our customers in the 2015 report. Our goal has been to reduce fear, uncertainty, and doubt around these events, and give our customers clear and accurate advice about which flaws impact the Red Hat technologies they use.

2016 marked the [15th](#) anniversary of Red Hat Product Security and saw a new layout for the report.



Figure 3. 2016 Red Hat Product Security risk report

Because we have been writing the report for so many years, we have been able to note many trends and patterns. We have seen [bleeding hearts](#), [dirty COWs](#), and [melting ghosts](#), and maintained clear, calm analysis and advice to help our users manage their risk. We have seen technology move from desktops to client-server, to private cloud, to multiple public clouds, and now we operate in an environment with open hybrid cloud, where workloads can move to locations around the world, shifting clouds and suppliers to meet the dynamic needs of modern business.

2021 will be the 20th year of the Red Hat Product Security team helping protect open source. Red Hat Product Security has grown to be one of the most respected voices within the product security community. Our developed partnerships with open source communities and closed source vendors have allowed us to serve our communities effectively, improve overall security, and reduce the risks associated with free and open source software.

We are proud to continue the tradition of openness and transparency with our 2020 Red Hat Product Security Risk Report. Now that we have discussed past reports that have led us here, we will review the 2020 data.

Sources and methodologies

Red Hat Product Security is a member of the [FIRST CVSS SIG](#) and uses the industry-standard [Common Vulnerability Scoring System \(CVSS\)](#) as an additional measurement on each vulnerability we address. All CVEs impacting Red Hat products are issued a CVSSv3.x score.

As we issue a true CVSS score for Red Hat software, we assume our products are used as designed, with security-focused defaults and settings in place.

While [not a measurement of risk](#), CVSS helps inform us precisely how a particular vulnerability works and what aspects of the information security triad—confidentiality, integrity, and availability (CIA)—the flaw impacts. Product Security uses CVSS as part of our holistic assessment of the vulnerability and how it impacts software in our portfolio. We also conduct other analyses such as developing reproducers, analyzing the impact to layered products, and looking at how the flaw measures against our build and compiling practices.

Ultimately, we use a [four-point scale](#) to objectively describe a particular bug's severity based on rigorous analysis of the flaw. We designed this scale to align closely with similar scales used

throughout the industry by other vendors and upstream open source communities. Our intent for the severity levels is to help users determine which issues could pose more risk. Ideally, this prioritized risk assessment helps customers understand how they are exposed and allows them to better schedule updates to the systems they manage. We recognize that each business is unique, with its own requirements and challenges, and that all risks are not created equal, nor are they the same company to company.

The four-point scale rates vulnerabilities as Low, Moderate, Important, or Critical. Critical vulnerabilities pose the most severe risk to an organization. As described in our rating methodology, a Critical vulnerability could be exploited remotely over a network (or the internet) or be automated in an attack, such as by a worm. Like many of our peers, we expand this definition to include flaws that affect web browsers or browser plug-ins that users might be susceptible to if they visit malicious or compromised websites.

When Red Hat Product Security reviews a flaw, we look at how the software is sourced, built, packaged, and deployed. As we issue a true CVSS-score for Red Hat software, we assume our products are used as designed, with security-focused defaults and settings in place. If changes are made to system settings or security controls outside of the baseline, sysadmins should take that into account as they evaluate the risk a vulnerability might pose inside their unique environments. It is important to remember that no outside vendor can tell a business what risks are important to them or what actions to take to protect their sensitive data. CVSS and the Red Hat Severity rating are baselines for our software consumers to begin their own risk assessment.

Red Hat releases advisories (Red Hat security advisory or RHSA) for each set of fixes for an impacted offering. These RHSAs are product/service-specific and can contain fixes for one or multiple CVEs. We share this data as a [web-based database](#), through a [mailing list](#), through data feeds like our [OVAL](#) data feed, [CVRP/CSAF](#), as [flat-files](#), or through our [Vulnerability Data API](#). Subscribers using our Red Hat [Insights](#) vulnerability managed service also get proactive notifications as new updates are published.

Vulnerabilities

Across all Red Hat offerings and for all issue severities, we fixed more than 2,000 vulnerabilities by releasing more than 1,500 security advisories in 2020. Looking back at previous years, we can see a marked increase in both CVEs and RHSAs. CVEs are a steady stream of work that needs to be addressed by the open source community, Red Hat, and customers. Ensuring that systems are up to date with the latest fixes is critical to the foundation of organizational security. Figure 4 shows that in the future, based on historical trends, we certainly should expect more, not fewer, fixes to be provided for the portfolio.

RHSAs, CVEs and Reported Flaws

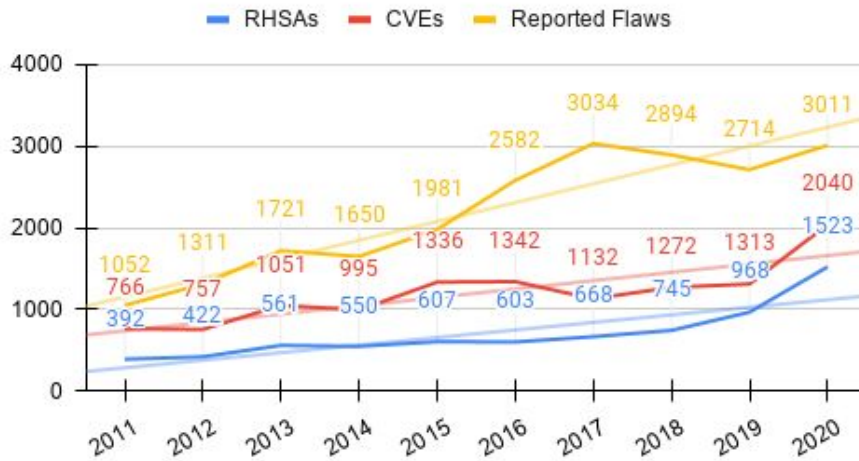


Figure 4. RHSAs, CVEs, and reported flaws

We issued 53 Critical security advisories that addressed 19 Critical CVEs. Interestingly, 31% of these Critical security advisories were issued within one day of the issue becoming public. We'll dive into the details as the report progresses, but an interesting fact to note is that of those 19 Critical CVEs, only 4 represent non-browser flaws. By managing the packages you deploy and not installing web browsers (whether represented in chromium, firefox, or the ever-favourite flash-plugin) you immediately avoid any of the threats and risks associated with those frequently-exploited tools.

Looking at the average delivery time for Critical advisories, we delivered the advisories within an average of 6 days of the issue becoming public, with the median being four days. It is important to point out that of these Critical CVEs, all but four exclusively impacted browser-based components. Of the four remaining, two were in .NET (dotnet), one in Haproxy, and the last in CloudForms. Here we attained 50% fixed within one business day and 100% within one week. We addressed a record number of 425 Important CVEs through 859 RHSAs during the same time period. For these vulnerabilities, 11% had initial patches available within one business day, with the average delivery time of 60 days and the median of 16 days. All three of these measurements (Flaws, CVEs, and Advisories) have progressively grown "up and to the right" throughout our time analyzing this information. We believe this trend will continue into the foreseeable future.

CVEs by severity - All offerings

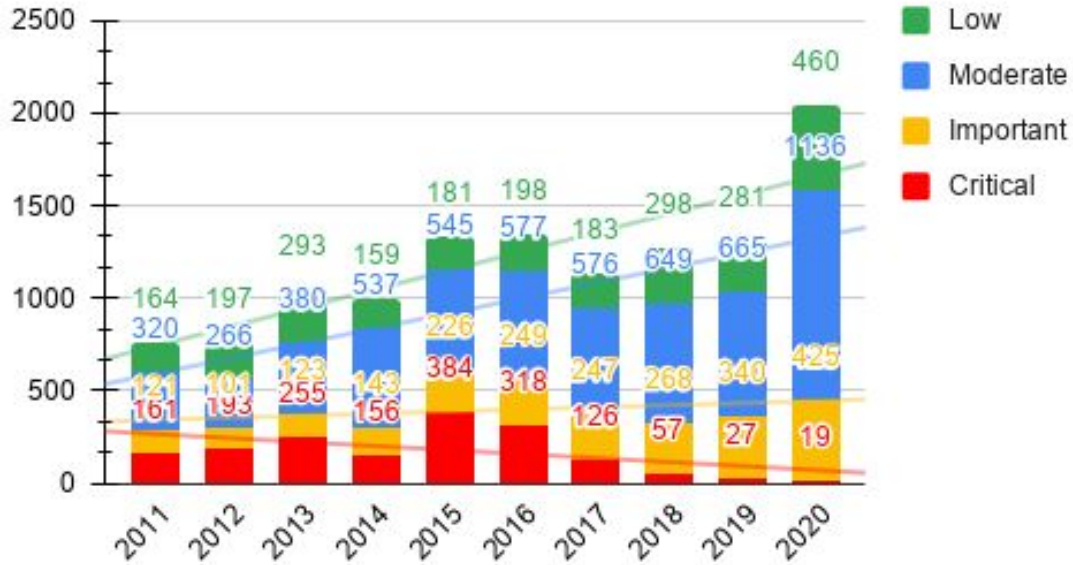


Figure 5. CVEs by severity

Critical, Important, Moderate, and Low CVEs - 2020

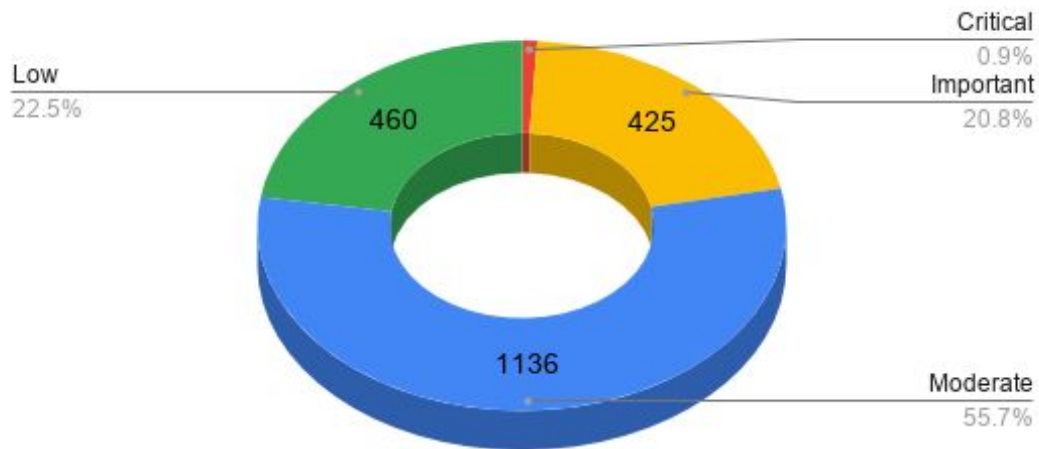


Figure 6. Critical, Important, Moderate, & Low CVEs—2020

The year at a glance

- 3,011 security issues were reported to Red Hat Product Security (slightly up from 2019).
- 2,040 CVEs were addressed throughout 2020, a 55% increase from 2019.
- 1,523 Red Hat security advisories were issued, a record increase over previous years.
- 53 Critical advisories addressed 19 Critical vulnerabilities.
- 31% of Critical issues were addressed within 1 business day—slightly less than last year.
- 89% of Critical issues were addressed within 1 week—similar to last year.
- 100% of Critical issues were addressed within 31 days of public release—exceeding 2019’s performance.
- 11% of Important issues were addressed within 1 business day—slightly down from last year.
- 28% of Important issues were addressed within 1 week—roughly on pace with 2019.
- 64% of important issues were addressed within 31 days—exceeding 2019’s delivery.

To contextualize the delivery statistics while comparing 2019 and 2020, 2020 saw an uptick of work coming in and delivered to subscribers. 2020 saw a volume increase of nearly 1.5 times over the previous year, which was the third highest volume year on record. So while the speed of delivery is slightly affected, more updates are being provided through our support streams. This result comes from a 2019 policy change in which our Red Hat Enterprise Linux line of business changed to [fix more vulnerabilities](#) by expanding our CVE coverage.

From 2019, we see that the frequency of Critical severity issues continues its downward trend to the lowest level we have seen on record. Conversely, Important and lower issues continue to rise across the portfolio. Important issues alone increased by 80%. Red Hat fixed more Moderate vulnerabilities than ever, correcting 1,136 Moderate CVEs. At a 70% increase, this is nearly as many Moderate CVEs as the previous two years combined. Finally, Red Hat released patches for 460 Low severity vulnerabilities, a 61% increase over the previous year.

The same CVE can have different effects depending on how the product is compiled or deployed. Even within a single product like Red Hat Enterprise Linux, there is potentially high variability for how a vulnerability can impact the supported release streams. Red Hat Product Engineering crafts a default deployment configuration that system administrators can alter by enabling or disabling features. Usually, not every package is installed, nor are some likely installed in an enterprise installation.

As software matures, historical practices are evolved or abandoned, and features are added or removed. Comparing vulnerabilities between versions of the Red Hat products yields interesting but not very useful comparisons beyond trends or reflective efforts put into building those offerings. The offerings are acutely representative of development practices used within

the open source community and will evolve and advance over time. Typically, high volumes of CVEs and bug fixes are addressed as new major versions of software are released.

The figure below compares the advisory counts of a subset of our Red Hat Enterprise Linux product family and others within our portfolio. A single Red Hat security advisory (RHSA) will often fix multiple vulnerabilities across multiple versions of a product. We view the vulnerability count as an indication of the amount of effort a customer will spend to both understand and then patch or mitigate the issue within their environment.

Red Hat security advisory (RHSA) chart—2020				
Product	Critical	Important	Moderate	Low
All	53 [^]	859 [^]	591 [^]	107 [^]
Red Hat Enterprise Linux 6, 7, 8	44 [^]	616 [^]	371 [^]	65 [^]
Red Hat JBoss® Enterprise Application Platform—all supported versions	3 [^]	64 [^]	6 ^v	0 ^v
Red Hat OpenShift® Container Platform—all supported versions	1 [^]	56 [^]	126 [^]	24 [^]
Red Hat OpenStack® Platform—all supported versions	0 ⁻⁻	20 ^v	18 ^v	2 ^v

Figure 7. RHSA comparison chart

Legend
 v = trend down
 ^ = trend up
 -- = no trend

Red Hat designs products to run best on other Red Hat products but to also work well with other operating systems. Our 25-year record of providing one of the most widely used operating systems, Red Hat Enterprise Linux has taught us a lot about managing open source software. Our solutions start with the foundation of Red Hat Enterprise Linux and build upon it. When looking at a Red Hat solution, keep in mind that each layer of the stack is updated separately. To get a holistic view of maintenance and updates, you must view multiple channels and repositories to ensure you are keeping current on each package and component you are using.

Layered products, like Red Hat OpenShift Container Platform, Red Hat OpenStack Platform have their own flaws that need to be tracked and addressed. The magnitude and volume of changes in these solutions are far smaller than the base operating system, which provides 1,348

packages for the default Red Hat Enterprise Linux 8.3 (with GUI) installation. System administrators and DevOps teams would not only need to think through the deployment of 207 updates for their OCP infrastructure but also the 1,096 updates released for Red Hat Enterprise Linux in 2020.

In Figure 8, the numbers reflect the default installations of those products. Red Hat delivers products in a generally secured state with reasonable, more secure defaults (intended to cover the maximum amount of reasonable business cases) and services enabled. Customers looking to reduce their threat footprint should consider additional hardening beyond the defaults, as detailed in documents like the [Red Hat Enterprise Linux 8 security hardening guide](#). The steps and techniques there help further protect systems. Along with that guidance, customers can install or remove packages and processes they do not need to reduce the potential threats they might be exposed to throughout the normal course of operations. It is worth noting that when default security features are disabled, like turning off SELinux, the risk profile of that system is drastically altered, opening up the potential for additional security risks and impacts.

Product	Number of packages
Red Hat Enterprise Linux 8.3 Server (minimal)	384 RPMs
Red Hat Enterprise Linux 8.3 Workstation (default w/GUI)	1,376 RPMs
Red Hat Enterprise Linux 7.9 Server (minimal)	343 RPMs
Red Hat Enterprise Linux 7.9 Workstation (default w/ GUI)	1,449 RPMs
Red Hat OpenStack Platform 16.1	767 RPMs + underlying OS
Red Hat OpenShift Container Platform 4.6	119 containers + underlying OS
Red Hat JBoss Enterprise Application Platform 7.3.4	1,132 jars + underlying OS
Red Hat Smart Management (on Red Hat Enterprise Linux 7)	434 core packages + 64 Smart Management packages + underlying OS

Figure 8. Red Hat offering package counts

How you manage your environment’s attack surface directly impacts your organization’s risk profile and volume of operational effort. The importance of this management cannot be stressed enough. Only install packages, libraries, and other software components that you require for the operation of your systems. Trimming the attack surface of a system down

through hardening techniques exposes fewer openings for a malicious attacker to exploit the system and ultimately creates less work for your operations team because they do not need to track, patch, and update as much software. As we have shown, modern software vulnerabilities have dramatically increased over the years, and it is important to reduce your organization's risk by avoiding unnecessary packages lurking on the system that must be maintained and secured continuously. Your security and operations teams will thank you for being a cautious steward of your systems.

Vulnerability trends

As a general rule, as software matures, fewer problems are found. Several factors contribute to this, including the notion that ideally, as the developers iterate the codebase over time, problems are identified and fixed. One of the biggest challenges to enterprises is the frequent churn in newer projects where features and functions rapidly change (sometimes taking completely different directions than the project initially started with). Along with these security updates we describe in our report, one of the greatest values of a Red Hat subscription is the stability, testing, and maturation that Red Hat's engineering processes bring to our solutions. We help reduce organizations' impacts through a technique called [backporting](#) while providing the innovation in which the open source ecosystem thrives. Backporting delivers precise bug fixes, features, or required security fixes but does not bring along the *beta* aspects of iterative development.

Average and median resolution - All offerings

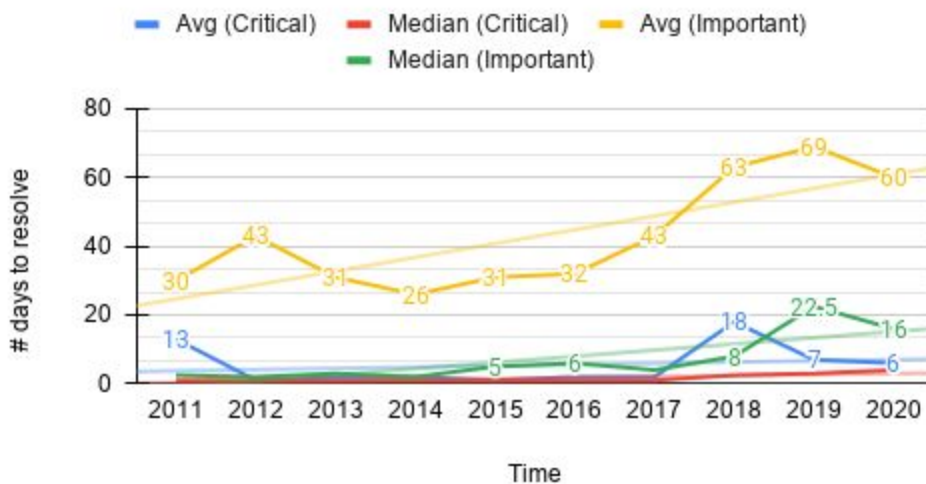


Figure 9. Average and median resolution

With the increase in volume and complexity of the software, there come some inevitable tradeoffs. For Critical severity issues, which represent the most likely and potentially most impactful vulnerabilities, Red Hat has kept the median (4 days) and average (6 days) days to resolve within seven business days, with 100% of Critical issues being addressed across the whole portfolio within 31 days of public disclosure. The same statistics for Important rated issues, which are close to 22 times the volume of Critical issues, were fixed within higher median (16 days) and average (60 days) timeframes with 64% of all Important CVEs being addressed within 31 days of public disclosure.

Putting the ecosystem in context, in 2015, Red Hat's portfolio consisted of around 100 streams of products and versions. In 2020, we oversaw over 150 unique products and versions, with four versions of Red Hat Enterprise Linux in some form of active [support or development](#) (which includes 16 unique kernel versions that needed support). One of the most important values our customers see out of their subscriptions is the stability and engineering that goes into the delivered products and services.

For example, Red Hat Enterprise Linux Engineering and Quality Engineering typically will spend 48-96 hours testing each release candidate erratum for the Linux kernel. This process fixes the issue and continues to function as expected and as performant as possible. Adding to Quality Engineering's burden has been the explosion of new layered products that depend upon that kernel. Each of these products needs consideration as lower-level changes are made. Just because it works as expected in Red Hat Enterprise Linux does not mean there is no *downstream* impact to OpenStack or Red Hat Data Services. This thorough, methodical testing helps minimize the risk of a newly secured package causing unintended surprises

In summary, delivery times have risen over the years due to the increase of CVEs, the number of packages covered by our support life cycles, and the expansion of Red Hat Enterprise Linux Extended Update Support Add-On.

Analyzing the metrics further, we will compare CVEs at the offering level. The doughnut chart in Figure 10 helps us see the relative volumes of each of our major product lines as percentages of the total CVE output.

CVEs by product family

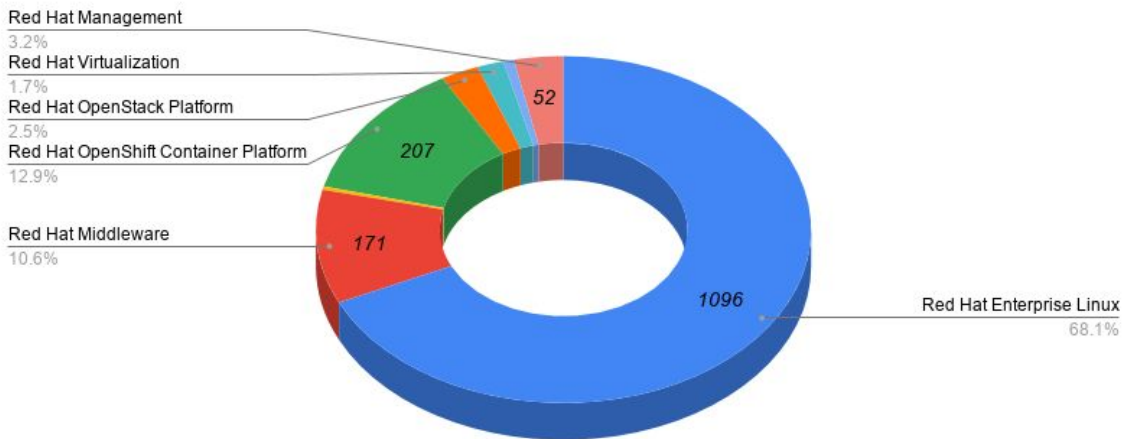


Figure 10. CVEs by product family

Red Hat Enterprise Linux tops the chart with the highest volume of fixes delivered. This ranking is for several reasons, including the number of components included within the offering and other layered solutions that sit on top of that stable base (OpenShift Container Platform, OpenStack Platform, JBoss EAP, etc.), which all rely on components curated within that foundation. These components and layered solutions are something that system administrators need to be aware of, as they maintain their fleets to ensure they address all potential vulnerabilities that may impact them.

CVEs by product family

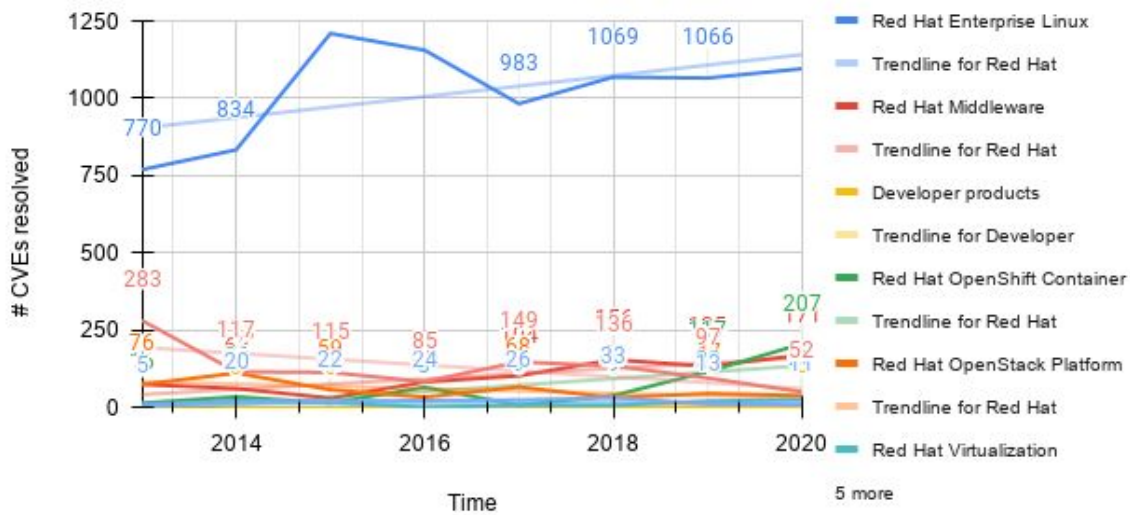


Figure 11. CVEs annually by product family

One of the greatest values of a Red Hat subscription is the stability, testing, and maturation that Red Hat’s engineering processes bring to our solutions. Red Hat Enterprise Linux Engineering and Quality Engineering typically will spend 48-96 hours testing each release candidate errata for the Linux kernel.

Looking at that data over a timeline, you can see Red Hat Enterprise Linux’s continued CVE prominence in the volume of issues reported and addressed. However, newer generations of offerings like OpenShift Container Platform increase their use and scope of what they depend upon are growing in volume as well.

To analyze the volume of CVEs a product might incur over its [lifetime](#), refer to Figure 12 below:

Red Hat Enterprise Linux major release CVE stats - All severities

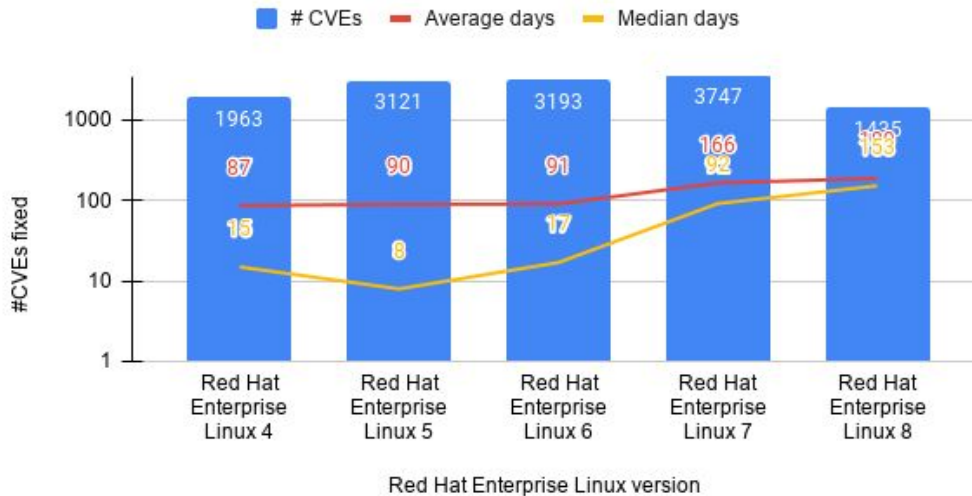


Figure 12. Red Hat Enterprise Linux major release CVE stats

At the time of the publication of this report, Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5 are retired and no longer receiving further errata. Over their lifetimes, they received fixes for 1,963 and 3,121 CVEs respectively. Red Hat Enterprise Linux 4 was released to the public on February 14, 2005 and went end-of-life (EoL) on March 31, 2017, with the retirement of Red Hat Enterprise Linux 4.9's End of Life Support (ELS) phase. Red Hat Enterprise Linux 5 came out March 15, 2007 and passed out of ELS on November 30, 2020 for 13 years of total service to our customers.

The figures of Red Hat Enterprise Linux 6 through Red Hat Enterprise Linux 8 are not as firm because all three versions are in various states of maturity in their support life cycles. You can see a trend that the more modern solutions are not only including more packages and components within but also fixing more CVEs than before. With Red Hat Enterprise Linux 7 continuing with Maintenance 2 support through 2024 and Red Hat Enterprise Linux 8 not reaching the end of that milestone until 2029, we fully expect to see an overall increase in volume in the number of issues addressed on those platforms.

We feel it is important to address the root cause of these vulnerabilities. For every vulnerability that impacts a component of our portfolio, Red Hat ensures it is tracked with a CVE identifier, issued a CVSS score, and gets a Red Hat severity rating. We also determine the coding problem behind the vulnerability. This issue is tracked using an industry standard called [CWE](#) (Common Weakness Enumeration), another excellent tool curated by MITRE. Figure 13 details the

components that had the highest number of CVEs fixed in 2020, along with all of the CWE identifiers describing the programmatic root cause of how those flaws came to be.

Component	Number of CVEs fixed	CWEs of the fixed CVEs
chromium-browser	246	CWE-416(26), CWE-358(6), CWE-122(5), CWE-125(4), CWE-843(3), CWE-20(3), CWE-476(3), CWE-190(2), CWE-203(1), CWE-120(1), CWE-787(1), CWE-190->CWE-122(1)
kernel	176	CWE-416(26), CWE-400(23), CWE-787(11), CWE-200(11), CWE-476(10), CWE-122(10), CWE-119(8), CWE-120(6), CWE-20(4), CWE-250(4), CWE-125(4), CWE-284(4), CWE-440(3), CWE-362(2), CWE-362->CWE-416(2), CWE-362->CWE-667(2), CWE-772->CWE-200(2), CWE-835(2), CWE-843(2), CWE-284->CWE-201(2), CWE-862(1), CWE-226(1), CWE-672(1), CWE-400->CWE-476(1), CWE-331->CWE-200(1), CWE-20->CWE-250(1), CWE-226->CWE-385->CWE-203(1), CWE-253->CWE-476(1), (CWE-190 CWE-125)(1), (CWE-400 CWE-122)(1), CWE-20->CWE-200(1), CWE-190(1), CWE-401->CWE-400(1), CWE-385->CWE-203(1), CWE-416->CWE-476(1), CWE-772(1), CWE-460(1), CWE-362->CWE-200(1), CWE-667(1), (CWE-787 CWE-119)(1), CWE-327(1), CWE-121(1), CWE-349(1), CWE-626(1), CWE-805(1), CWE-20->CWE-119(1), CWE-20->CWE-476(1), CWE-1251(1), CWE-20->CWE-835(1), CWE-401->CWE-416(1), CWE-94(1), CWE-248(1), CWE-131(1), CWE-787->CWE-250(1), CWE-319(1), CWE-732(1), CWE-367(1)
kernel-rt	147	CWE-416(22), CWE-400(22), CWE-787(10), CWE-119(8), CWE-200(8), CWE-122(7), CWE-476(7), CWE-120(5), CWE-125(4), CWE-250(3), CWE-20(3), CWE-440(3), CWE-362->CWE-416(2), CWE-362->CWE-667(2), CWE-772->CWE-200(2), CWE-284(2), CWE-843(2), CWE-284->CWE-201(2), CWE-862(1), CWE-672(1), CWE-400->CWE-476(1), CWE-331->CWE-200(1), CWE-20->CWE-250(1), CWE-253->CWE-476(1), (CWE-190 CWE-125)(1), (CWE-400 CWE-122)(1), CWE-190(1), CWE-401->CWE-400(1), CWE-385->CWE-203(1), CWE-416->CWE-476(1), CWE-362(1), CWE-772(1), CWE-460(1), CWE-362->CWE-200(1), CWE-667(1), (CWE-787 CWE-119)(1), CWE-121(1), CWE-349(1), CWE-805(1), CWE-20->CWE-119(1), CWE-20->CWE-476(1), CWE-20->CWE-835(1), CWE-401->CWE-416(1), CWE-94(1), CWE-248(1), CWE-131(1), CWE-787->CWE-250(1),

		CWE-319(1), CWE-732(1), CWE-367(1), CWE-835(1)
webkitgtk4	101	CWE-94(8), CWE-119(2), CWE-416(2), CWE-200(1), CWE-20(1), CWE-79(1), CWE-400(1)
mysql:8.0	99	CWE-400(3)
rh-mysql80-mysql	99	CWE-400(3)
firefox	75	CWE-416(18), CWE-120(15), CWE-79(6), CWE-843(4), CWE-200(4), CWE-451(4), CWE-125(3), CWE-119(3), CWE-552(1), CWE-296(1), CWE-190(1), CWE-209(1), CWE-276(1), CWE-138(1), CWE-648(1), CWE-601(1), CWE-829(1), CWE-354(1), CWE-358(1), CWE-212(1), CWE-20(1), CWE-787(1), CWE-20->CWE-78(1)
thunderbird	74	CWE-416(18), CWE-120(15), CWE-79(6), CWE-200(6), CWE-843(3), CWE-125(3), CWE-119(3), CWE-552(1), CWE-172(1), CWE-296(1), CWE-209(1), CWE-648(1), CWE-601(1), CWE-829(1), CWE-354(1), CWE-451(1), CWE-358(1), CWE-212(1), CWE-121(1), CWE-20(1), CWE-456(1), CWE-312(1), CWE-476(1), CWE-20->CWE-78(1)
ImageMagick	73	CWE-400(22), CWE-772(9), CWE-835(5), CWE-119(5), CWE-125(5), CWE-122(3), CWE-416(3), CWE-456(3), CWE-617(2), CWE-401(2), CWE-476(2), CWE-193(2), CWE-369(2), CWE-20->CWE-400(1), CWE-125->CWE-200(1), CWE-248(1), CWE-200(1), CWE-119->CWE-122(1), CWE-787(1), CWE-121(1)
kernel-alt	58	CWE-416(12), CWE-200(7), CWE-122(6), CWE-400(6), CWE-787(3), CWE-476(2), CWE-120(2), CWE-119(2), (CWE-400 CWE-122)(1), CWE-20->CWE-200(1), CWE-20(1), CWE-190(1), CWE-401->CWE-400(1), CWE-772(1), CWE-440(1), (CWE-787 CWE-119)(1), CWE-362->CWE-416(1), CWE-843(1), CWE-284->CWE-201(1), CWE-20->CWE-835(1), CWE-248(1), CWE-367(1), CWE-835(1), CWE-121(1), CWE-125(1)
webkit2gtk3	51	CWE-119(4), CWE-416(3), CWE-841(2), CWE-841->CWE-79(2), CWE-400(1), CWE-20->CWE-119(1), CWE-20->CWE-79(1), CWE-77(1), CWE-119->CWE-416(1), CWE-20->CWE-125(1), CWE-284(1)
java-1.8.0-ibm	42	CWE-248(11), CWE-770(5), CWE-20(4), CWE-119(3), CWE-79(2), CWE-285(1), CWE-200(1), CWE-522(1), CWE-476(1), CWE-190(1), CWE-172(1), CWE-327(1), CWE-471(1), CWE-113(1), CWE-185->CWE-400(1)
java-11-openjdk	34	CWE-248(5), CWE-770(4), CWE-20(4), CWE-119(3), CWE-327(2), CWE-358(2), CWE-319(1), CWE-295(1),

		CWE-190(1), CWE-367(1), CWE-172(1), CWE-471(1), CWE-841(1), CWE-113(1), CWE-185->CWE-400(1)
java-1.8.0-openjdk	31	CWE-248(7), CWE-20(4), CWE-770(4), CWE-119(3), CWE-319(1), CWE-295(1), CWE-190(1), CWE-367(1), CWE-172(1), CWE-327(1), CWE-471(1), CWE-113(1), CWE-185->CWE-400(1)

Figure 13. Top CWEs by component

Last year’s package with the most CVEs—the Linux kernel as curated in our product set –plummets in the ranking this year, dropping to have only the ninth most CVEs addressed (176).

We have held the long-time opinion that giving a vulnerability a name, a logo, a theme song, and selling merchandise about it does not make that vulnerability intrinsically important. Our goal has been to reduce fear, uncertainty, and doubt around these events and to give our customers clear and accurate advice about which flaws impact the Red Hat technologies they use.

The chromium browser, with 246 CVEs for the year, came in first with the most CVEs in 2020. Chromium had CVEs related to [CWE-416](#) (use after free) 24 times over the course of 2020. This issue is a fairly common coding mistake where the developer references memory that had been recently cleared out, causing the program to crash.

Another alarming aspect of memory referencing errors is that it could lead to inadvertent exposure of unintended data or allow a malicious attacker to execute code on the affected system. Another set of packages to strongly scrutinize deploying would be the Java™ family of packages (openjdk/java et.

al.). These are incredibly powerful components that enable other higher-level operations, but do they need to be deployed on systems that do not require it? [CWE-248](#) (uncaught exception) occurs pretty frequently and could have severe consequences if exploited. An exception is an error in which the program could not find data or encountered some problem somewhere. The developer did not predict, plan for, test a scenario, and write logic into the program to see the error and react to it properly. Depending on the version of OpenJDK you are using, you may have had to fix four or more CVEs around just that one particular problem.

Let us highlight the need for the appropriate curation of your infrastructure again. All combined, if you have desktop components installed on your Red Hat Enterprise Linux servers, you were exposed to nearly 500 more CVEs through tools like the browser and email client.

Analyzing the CWE data helps Red Hat Product Security provide our developers and the broader community feedback on coding patterns to avoid and help a sysadmin and security practitioner understand the root coding problems that created that vulnerable condition. We recently wrote an article [exploring the security technologies of Red Hat Enterprise Linux](#) in

which the team explored CWEs more in-depth and found how the CWEs related to those controls. The article can benefit system administrators and anyone related to development practices within their organization.

Which issues were branded and which really mattered in 2020

We have held the long-time opinion that giving a vulnerability a name, a logo, a theme song, and selling merchandise about it does not make that vulnerability intrinsically important. Beginning in 2014, [CVE-2014-0160](#), also known as OpenSSL's "Heartbleed," brought marketing to the world of information security. Six years and dozens of logos later, the value of branding a flaw to raise awareness is still much debated within the industry.

To help assist our customers to both understand and react to these front-page style events, Red Hat created the [Customer Security Awareness program](#) (CSAw). Over the years, through the augmented pages of our [security bulletin](#) pages, we have shared information about these flaws, branded or not, to prepare our subscribers and the larger community. When a vulnerability reaches the level of a CSAw, Red Hat contributes to a more in-depth, richer experience that helps identify, educate, and contextualize these types of problems for the entire spectrum of personas that might read them.

In 2020, four issues have raised to this level that we will explore using Figure 13 as our reference point:

CVE	Name	Severity
CVE-2016-8867, CVE-2020-14298, CVE-2020-14300	Runc regression - docker-1.13.1-108	IMPORTANT
CVE-2020-10713	Boot Hole Vulnerability - GRUB 2 boot loader	MODERATE
CVE-2020-11100	haproxy: malformed HTTP/2 requests can lead to out-of-bounds writes	CRITICAL
CVE-2020-12351, CVE-2020-12352, CVE-2020-24490, CVE-2020-25661 & CVE-2020-2566	Bleeding Tooth	IMPORTANT & MODERATE

Figure 14. Customer security awareness events—2020

[Haproxy: malformed HTTP/2 requests \(2April2020\) CVE-2020-11100](#)

Severity Rating: **Critical** **CVSSv3 Score:** **8.8** **CWE:** **CWE-20->CWE-787**: Improper input validation leads to out-of-bounds write

When discovered, this problem affected our three platform offerings (Red Hat Enterprise Linux, Red Hat OpenShift Container Platform, and Red Hat OpenStack Platform). The flaw affected HAProxy (a reverse proxy and load-balancing component heavily used in our offerings to implement high availability) that unfortunately allowed the daemon to process certain malicious HTTP/2 request packets. This flaw allows an attacker to send crafted HTTP/2 request packets that cause memory corruption, leading to a crash or remote arbitrary code execution with the user's permissions running HAProxy. Red Hat released both updates to affected versions and guidance on how to mitigate the issue for enterprises seeking a temporary solution prior to releasing those updates.

[Runc regression in docker-1.13.1-108 \(23June2020\) CVE-2016-8867, CVE-2020-14298, & CVE-2020-14300](#)

Severity Rating: **Important** **CVSSv3 Score:** **7.5, 8.8, & 8.8** **CWE:** **CWE-271**: Privilege dropping / lowering errors

Red Hat released a version of docker for Red Hat Enterprise Linux 7 extras that introduced multiple regressions for previously fixed security flaws and a new vulnerability. Red Hat released this version in early January 2020 and subsequently fixed it with an early February 2020 release. Red Hat created advisories and a CVE to notify any end users using the older, out-of-date January files. This regression potentially impacted users of the extras package for an assortment of layered products by possibly allowing a malicious or compromised container to compromise the container host and other containers running on the same host.

[Boot Hole - GRUB2 boot loader \(29July2020\) CVE-2020-1073](#)

Severity Rating: **Moderate** **CVSSv3 Score:** **8.2** **CWE:** **CWE-787->CWE-78**: Out-of-bounds write leads to improper neutralization of special elements used in an OS command ('OS command injection')

We ended up with two branded vulnerabilities in 2020. This flaw allowed a resident attacker who had already gained access to have the ability to hijack the boot process and execute malicious code during system startup. Systems using UEFI secure boot, which protects systems by

verifying the software used to boot up a computer, can also be bypassed using this vulnerability. Hardening the Red Hat Enterprise Linux 8 kernel and associated processes helped stop the execution of this flaw on newer systems and reduced the need to patch the flaw for many users. Red Hat Enterprise Linux 7 was released before these newer techniques and required more extensive updates to the shim process to ensure this vulnerability was closed. This issue impacted Red Hat Enterprise Linux 7 and 8 along with OpenShift Container Platform and Red Hat Enterprise Linux CoreOS.

[Bleeding Tooth](#) (14Oct2020) [CVE-2020-12351](#), [CVE-2020-12352](#), [CVE-2020-24490](#), [CVE-2020-25661](#) & [CVE-2020-25662](#)

Severity Rating: Important and Moderate **CVSSv3 Score:** 8.8, 5.3, 7.1, 8.8, & 5.3 **CWE:** **CWE-843:** Access of resource using incompatible type ('type confusion'), **CWE-284->CWE-201:** Improper access control leads to insertion of sensitive information into sent data, **CWE-122:** Heap-based buffer overflow

The second branded flaw that directly impacted Red Hat offerings came to us through Bluetooth. These flaws allow a remote attacker within Bluetooth range to perform a system crash, execute arbitrary code, or leak small portions of stack memory from the system. Due to Red Hat getting included in the predisclosure efforts, we were unable to deliver the required fixes to our Red Hat Enterprise Linux 8.3 general announcement release, which was in the final stages of release preparation, ready for distribution at the time. We had to issue two additional errata to ensure users of that stream received the necessary fixes to correct this regression. In looking across our portfolio's primary consumers, the majority of our user base would not have been impacted by the potential threat, because servers and cloud computing and container resources rarely, if ever, use Bluetooth. The issue incurred a lot of media attention as we opted to manage it as part of our CSAw process to ensure our users were educated about the real scope and impact of the vulnerabilities.

Turning away from critical incidents, let us review what types of vulnerabilities our subscribers and the internet found interesting this year:

CVE	Severity	Title	Page views
CVE-2020-1938	Important	tomcat: Apache Tomcat AJP File Read/Inclusion Vulnerability	20,547
CVE-2020-10713	Moderate	grub2: Crafted grub.cfg file can lead to arbitrary code execution during boot process	17,464
CVE-2020-8617	Important	bind: A logic error in code which checks TSIG validity can be used to trigger an assertion failure in tsig.c	16,084
CVE-2020-8177	Moderate	curl: Incorrect argument check can allow remote servers to overwrite local files	15,019
CVE-2016-2183	Moderate	SSL/TLS: Birthday attack against 64-bit block ciphers (SWEET32)	11,295
CVE-2020-1971	Important	openssl: EDIPARTYNAME NULL pointer de-reference	11,257
CVE-2019-14287	Important	sudo: Privilege escalation via 'Runas' specification with 'ALL' keyword	10,172
CVE-2020-8622	Moderate	bind: truncated TSIG response can lead to an assertion failure	9,559
CVE-2018-15473	Low	openssh: User enumeration via malformed packets in authentication requests	9,486
CVE-2019-18634	Important	sudo: Stack based buffer overflow when pwfeedback is enabled	8,811

Figure 15. Most interesting CVE page views

2020 Most Interesting CVEs list

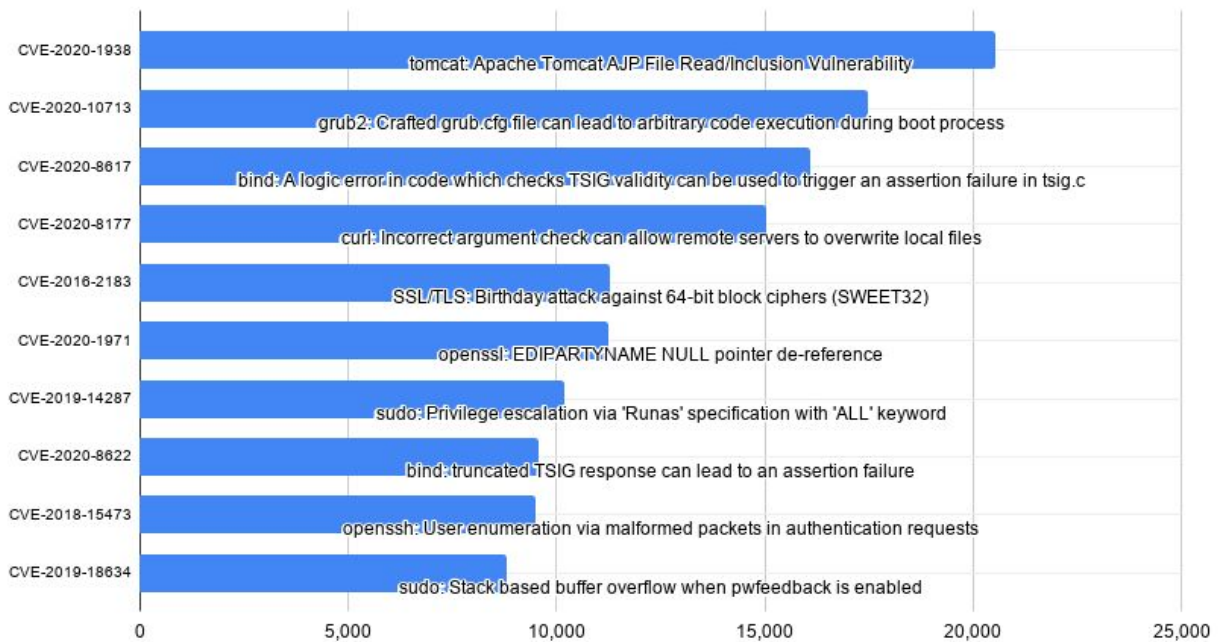


Figure 16. Most interesting CVEs

None of the CSAw events rose to a high level of concern (as measured by CVE page views). We can speculate that fewer customers were actually impacted by our *big four* this year, so the technical pages got less usage. Our readers widely viewed articles regarding the constant list-makers, bind, tomcat, sudo, kernel, and OpenSSH. These packages run core processes within our software and within our customers' mission-critical applications. They tend to end up widely touched on for more information by curious or affected administrators every year. While not as flashy as 2018's Spectre and Meltdown, which earned over 125,000 page views.

The open source ecosystem

Red Hat is an enterprise software company that uses an open source development model to compose its offerings. Every line of code derives from upstream open sources, and ultimately, any changes we make are provided publicly for downstream consumers of the software. We have a strong ethos of working community-first to ensure development is done in the open and takes advantage of all of the creativity and expertise of the global community of developers. Each Red Hat-branded solution comprises hundreds to thousands of open source packages

and projects, as Figure 7 illustrated. These communities provide vital innovation that becomes the foundation for our products and services.

Red Hat Product Security's mission is twofold:

- Help oversee the productization of these solutions that will ultimately end up in companies around the world, ensuring they are composed, managed, and delivered more securely.
- Monitor the components of the solutions we provide, and as vulnerabilities are discovered in those pieces, to document and describe them and work with Product Engineering to address them appropriately.

We fulfill these dual missions by working closely with internal Red Hat teams (Product Engineering, Quality Engineering, Support, etc.), the broader open source community, external peers, and security researchers.

In 2020, Red Hat Product Security investigated 3,011 vulnerabilities that potentially affected components of our offerings. This extensive list ultimately determined that 2,040 of these reports were vulnerabilities that affected Red Hat products, where we needed to take some action. These reports were recorded in our public Bugzilla system and shared externally once any embargoes were over. Each issue that impacts our products is assigned a CVE, a Red Hat severity score, and a CVSS score. All of this data is available through multiple streams for anyone to review:

- [Metrics webpage](#)
- [Red Hat security vulnerability data API](#)
- [OVAL](#) and [CVRF](#) data feeds
- [RHSA announcements](#)

We use this data to create metrics and review trends with Product Engineering to improve future releases and the entire open source ecosystem.

Red Hat does not sit idly by waiting for problems to fall into our laps; we are also proactively engaged internally and externally, seeking problems that could affect our offerings and subscribers. Approximately 30% of Critical issues Red Hat addressed came to us directly from peers, Red Hat employees, or Red Hat customers. This is slightly up from 2019, but the number of total flaws reported grew 11% over the previous year. Whenever possible, we share these issues with upstream and our industry peers. In addition to those issues, Red Hat may also find and report flaws in software that are not part of our currently shipped products. When it comes to fixing issues in third-party software, relationships matter. Red Hat Product Security and Product Engineering have deep ties into upstream and the technology industry at large. We are constantly communicating and collaborating with our peers on issues that impact all of our shared customers and communities.

If an upstream community is willing to share information about flaws with us in advance, we feel responsible for giving value back for that shared trust. We do this by reviewing advisories, checking patches, and feeding data back from our quality or performance testing groups. Ultimately we are focused on providing remediation to the flaws, and we all try to contribute positively to the solution as it is evolving.

Conclusion

2020 was a year full of unimaginable circumstances and security challenges across the technical spectrum. We hope that the review of the vulnerabilities that impacted our little corner of the world over the last calendar year was educational and enlightening. Our goal is to provide clear, calm analysis and recommendations on dealing with these threats and vulnerabilities, and to better manage your cybersecurity risk.

Risk is a different topic for every organization, a topic we've discussed through our "[Security in the modern IT world](#)" blog series and will continue to be as long as tough choices present themselves through security threats and vulnerabilities.

No vendor can accurately tell you what your organization's risks are based upon the following:

- They do not have insights into where your more critical business data resides.
- They do not have access to your long- and short-term strategic plans.
- They do not have insights into your customer bases, and they do not understand how an incident can impact your customer sentiment or what your regulators/assessors will see as a well or poorly managed process.

Risk manifests itself differently based on the data, system, or personnel in danger.

In general, everyone wants to eliminate vulnerabilities. Your organization may be most concerned about software flaws in an online or cloud context, while others may be concerned about IoT and other devices within the bounds of their private networks. Risk is all about context. That is the intention of this report and our daily operations within Red Hat Product Security—we strive to contextualize the vulnerabilities within the context of facts and data and how they function within the ecosystem of our software offerings. Receiving and understanding that data helps you to have a solid vulnerability management program that allows you to know what threats could potentially impact you, where your most important issues are, and where and when you have to react most quickly.

No control is perfect and protects against every vulnerability. Think holistically about your security controls (your tools, your processes, the training you provide your staff and



customers). Have layered or overlapping protections in place. Know where your critical data and systems are, and focus your security efforts there for maximum effectiveness. Ideally, you will stop all of your attackers, but if not, these blended controls should alert you of some maleficence and allow you to react quickly and prevent an incident before it gets out of your control.

We are very proud to have been a catalyst within our communities and the broader industry, helping contextualize security for nearly 20 years. As we showed this report's evolution over the last 14 years, we have always strived to describe the most impactful threats to our stakeholders. For the latest in what is going on with security for Red Hat products and services, please see the [Red Hat Product Security Center](#) to connect to all the security information within our portfolio.