A walk through Project Zero metrics

googleprojectzero.blogspot.com/2022/02/a-walk-through-project-zero-metrics.html

Posted by Ryan Schoen, Project Zero

tl;dr

- In 2021, vendors took an average of 52 days to fix security vulnerabilities reported from Project Zero. This is a significant acceleration from an average of about 80 days 3 years ago.
- In addition to the average now being well below the 90-day deadline, we have also seen a dropoff in vendors missing the deadline (or the additional 14-day grace period). In 2021, only one bug exceeded its fix deadline, though 14% of bugs required the grace period.
- Differences in the amount of time it takes a vendor/product to ship a fix to users reflects their product design, development practices, update cadence, and general processes towards security reports. We hope that this comparison can showcase best practices, and encourage vendors to experiment with new policies.
- This data aggregation and analysis is relatively new for Project Zero, but we hope to do it more in the future. We encourage all vendors to consider publishing aggregate data on their time-to-fix and time-to-patch for externally reported vulnerabilities, as well as more data sharing and transparency in general.

Overview

For nearly ten years, Google's Project Zero has been working to make it more difficult for bad actors to find and exploit security vulnerabilities, significantly improving the security of the Internet for everyone. In that time, we have partnered with folks across industry to transform the way organizations prioritize and approach fixing security vulnerabilities and updating people's software.

To help contextualize the shifts we are seeing the ecosystem make, we looked back at the set of vulnerabilities Project Zero has been reporting, how a range of vendors have been responding to them, and then attempted to identify trends in this data, such as how the industry as a whole is patching vulnerabilities faster.

For this post, we look at fixed bugs that were reported between January 2019 and December 2021 (2019 is the year we made changes to our disclosure policies and also began recording more detailed metrics on our reported bugs). The data we'll be referencing is publicly available on the Project Zero Bug Tracker, and on various open source project repositories (in the case of the data used below to track the timeline of open-source browser bugs).

There are a number of caveats with our data, the largest being that we'll be looking at a small number of samples, so differences in numbers may or may not be statistically significant. Also, the direction of Project Zero's research is almost entirely influenced by the choices of individual researchers, so changes in our research targets could shift metrics as much as changes in vendor behaviors could. As much as possible, this post is designed to be an objective presentation of the data, with additional subjective analysis included at the end.

The data!

Between 2019 and 2021, Project Zero reported 376 issues to vendors under our standard 90-day deadline. 351 (93.4%) of these bugs have been fixed, while 14 (3.7%) have been marked as WontFix by the vendors. 11 (2.9%) other bugs remain unfixed, though at the time of this writing 8 have passed their deadline to be fixed; the remaining 3 are still within their deadline to be fixed. Most of the vulnerabilities are clustered around a few vendors, with 96 bugs (26%) being reported to Microsoft, 85 (23%) to Apple, and 60 (16%) to Google.

Deadline adherence

Once a vendor receives a bug report under our standard deadline, they have 90 days to fix it and ship a patched version to the public. The vendor can also request a 14-day grace period if the vendor confirms they plan to release the fix by the end of that total 104-day window.

In this section, we'll be taking a look at how often vendors are able to hit these deadlines. The table below includes all bugs that have been reported to the vendor under the 90-day deadline since January 2019 and have since been fixed, for vendors with the most bug reports in the window.

Deadline adherence and fix time 2019-2021, by bug report volume

Vendor	Total bugs	Fixed by day 90	Fixed during grace period	Exceeded deadline & grace period	Avg days to fix
Apple	84	73 (87%)	7 (8%)	4 (5%)	69
Microsoft	80	61 (76%)	15 (19%)	4 (5%)	83
Google	56	53 (95%)	2 (4%)	1 (2%)	44
Linux	25	24 (96%)	0 (0%)	1 (4%)	25

Adobe	19	15 (79%)	4 (21%)	0 (0%)	65
Mozilla	10	9 (90%)	1 (10%)	0 (0%)	46
Samsung	10	8 (80%)	2 (20%)	0 (0%)	72
Oracle	7	3 (43%)	0 (0%)	4 (57%)	109
Others*	55	48 (87%)	3 (5%)	4 (7%)	44
TOTAL	346	294 (84%)	34 (10%)	18 (5%)	61

* For completeness, the vendors included in the "Others" bucket are Apache, ASWF, Avast, AWS, c-ares, Canonical, F5, Facebook, git, Github, glibc, gnupg, gnutls, gstreamer, haproxy, Hashicorp, insidesecure, Intel, Kubernetes, libseccomp, libx264, Logmein, Node.js, opencontainers, QT, Qualcomm, RedHat, Reliance, SCTPLabs, Signal, systemd, Tencent, Tor, udisks, usrsctp, Vandyke, VietTel, webrtc, and Zoom.

Overall, the data show that almost all of the big vendors here are coming in under 90 days, on average. The bulk of fixes during a grace period come from Apple and Microsoft (22 out of 34 total).

Vendors have exceeded the deadline and grace period about 5% of the time over this period. In this slice, Oracle has exceeded at the highest rate, but admittedly with a relatively small sample size of only about 7 bugs. The next-highest rate is Microsoft, having exceeded 4 of their 80 deadlines.

Average number of days to fix bugs across all vendors is 61 days. Zooming in on just that stat, we can break it out by year:

Bug fix time 2019-2021, by bug report volume

Vendor	Bugs in 2019 (avg days to fix)	Bugs in 2020 (avg days to fix)	Bugs in 2021 (avg days to fix)
Apple	61 (71)	13 (63)	11 (64)
Microsoft	46 (85)	18 (87)	16 (76)

Google	26 (49)	13 (22)	17 (53)
Linux	12 (32)	8 (22)	5 (15)
Others*	54 (63)	35 (54)	14 (29)
TOTAL	199 (67)	87 (54)	63 (52)

* For completeness, the vendors included in the "Others" bucket are Adobe, Apache, ASWF, Avast, AWS, c-ares, Canonical, F5, Facebook, git, Github, glibc, gnupg, gnutls, gstreamer, haproxy, Hashicorp, insidesecure, Intel, Kubernetes, libseccomp, libx264, Logmein, Mozilla, Node.js, opencontainers, Oracle, QT, Qualcomm, RedHat, Reliance, Samsung, SCTPLabs, Signal, systemd, Tencent, Tor, udisks, usrsctp, Vandyke, VietTel, webrtc, and Zoom.

From this, we can see a few things: first of all, the overall time to fix has consistently been decreasing, but most significantly between 2019 and 2020. Microsoft, Apple, and Linux overall have reduced their time to fix during the period, whereas Google sped up in 2020 before slowing down again in 2021. Perhaps most impressively, the others not represented on the chart have collectively cut their time to fix in more than half, though it's possible this represents a change in research targets rather than a change in practices for any particular vendor.

Finally, focusing on just 2021, we see:

- Only 1 deadline exceeded, versus an average of 9 per year in the other two years
- The grace period used 9 times (notably with half being by Microsoft), versus the slightly lower average of 12.5 in the other years

Mobile phones

Since the products in the previous table span a range of types (desktop operating systems, mobile operating systems, browsers), we can also focus on a particular, hopefully more apples-to-apples comparison: mobile phone operating systems.

Vendor	Total bugs	Avg fix time
iOS	76	70
Android (Samsung)	10	72

Android (Pixel)	6	72

The first thing to note is that it appears that iOS received remarkably more bug reports from Project Zero than any flavor of Android did during this time period, but rather than an imbalance in research target selection, this is more a reflection of how Apple ships software. Security updates for "apps" such as iMessage, Facetime, and Safari/WebKit are all shipped as part of the OS updates, so we include those in the analysis of the operating system. On the other hand, security updates for standalone apps on Android happen through the Google Play Store, so they are not included here in this analysis.

Despite that, all three vendors have an extraordinarily similar average time to fix. With the data we have available, it's hard to determine how much time is spent on each part of the vulnerability lifecycle (e.g. triage, patch authoring, testing, etc). However, open-source products do provide a window into where time is spent.

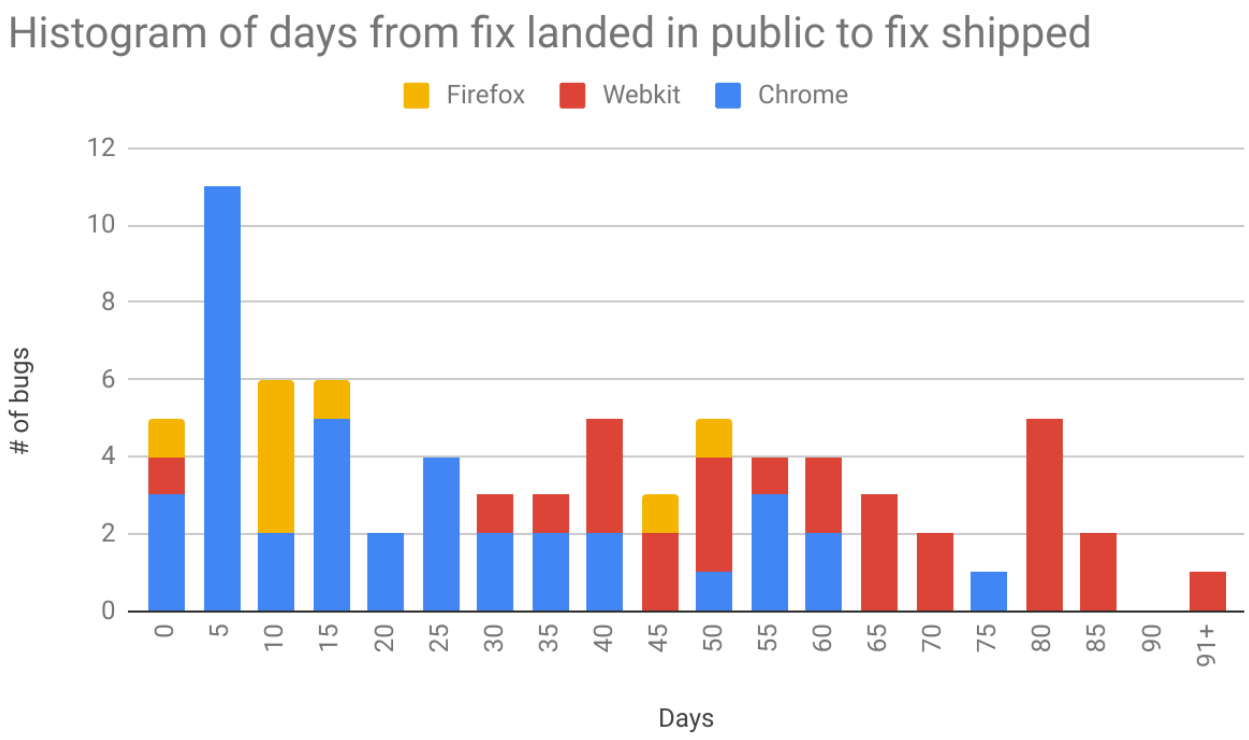
Browsers

For most software, we aren't able to dig into specifics of the timeline. Specifically: after a vendor receives a report of a security issue, how much of the "time to fix" is spent between the bug report and landing the fix, and how much time is spent between landing that fix and releasing a build with the fix? The one window we do have is into open-source software, and specific to the type of vulnerability research that Project Zero does, open-source browsers.

Fix time analysis for open-source browsers, by bug volume

Browser	Bugs	Avg days from bug report to public patch	Avg days from public patch to release	Avg days from bug report to release
Chrome	40	5.3	24.6	29.9
WebKit	27	11.6	61.1	72.7
Firefox	8	16.6	21.1	37.8
Total	75	8.8	37.3	46.1

We can also take a look at the same data, but with each bug spread out in a histogram. In particular, the histogram of the amount of time from a fix being landed in public to that fix being shipped to users shows a clear story (in the table above, this corresponds to "Avg days from public patch to release" column:



The table and chart together tell us a few things:

Chrome is currently the fastest of the three browsers, with time from bug report to releasing a fix in the stable channel in 30 days. The time to patch is very fast here, with just an average of 5 days between the bug report and the patch landing in public. The time for that patch to be released to the public is the bulk of the overall time window, though overall we still see the Chrome (blue) bars of the histogram toward the left side of the histogram. (Important note: despite being housed within the same company, Project Zero follows the same policies and procedures with Chrome that an external security researcher would follow. More information on that is available in our Vulnerability Disclosure FAQ.)

Firefox comes in second in this analysis, though with a relatively small number of data points to analyze. Firefox releases a fix on average in 38 days. A little under half of that is time for the fix to land in public, though it's important to note that Firefox intentionally delays committing security patches to reduce the amount of exposure before the fix is released. Once the patch has been made public, it releases the fixed build on average a few days faster than Chrome – with the vast majority of the fixes shipping 10-15 days after their public patch.

WebKit is the outlier in this analysis, with the longest number of days to release a patch at 73 days. Their time to land the fix publicly is in the middle between Chrome and Firefox, but unfortunately this leaves a very long amount of time for opportunistic attackers to find the

patch and exploit it prior to the fix being made available to users. This can be seen by the Apple (red) bars of the second histogram mostly being on the right side of the graph, and every one of them except one being past the 30-day mark.

Analysis, hopes, and dreams

Overall, we see a number of promising trends emerging from the data. Vendors are fixing almost all of the bugs that they receive, and they generally do it within the 90-day deadline plus the 14-day grace period when needed. Over the past three years vendors have, for the most part, accelerated their patch effectively reducing the overall average time to fix to about 52 days. In 2021, there was only one 90-day deadline exceeded. We suspect that this trend may be due to the fact that responsible disclosure policies have become the de-facto standard in the industry, and vendors are more equipped to react rapidly to reports with differing deadlines. We also suspect that vendors have learned best practices from each other, as there has been increasing transparency in the industry.

One important caveat: we are aware that reports from Project Zero may be outliers compared to other bug reports, in that they may receive faster action as there is a tangible risk of public disclosure (as the team will disclose if deadline conditions are not met) and Project Zero is a trusted source of reliable bug reports. We encourage vendors to release metrics, even if they are high level, to give a better overall picture of how quickly security issues are being fixed across the industry, and continue to encourage other security researchers to share their experiences.

For Google, and in particular Chrome, we suspect that the quick turnaround time on security bugs is in part due to their rapid release cycle, as well as their additional stable releases for security updates. We're encouraged by Chrome's recent switch from a 6-week release cycle to a 4-week release cycle. On the Android side, we see the Pixel variant of Android releasing fixes about on par with the Samsung variants as well as iOS. Even so, we encourage the Android team to look for additional ways to speed up the application of security updates and push that segment of the industry further.

For Apple, we're pleased with the acceleration of patches landing, as well as the recent lack of use of grace periods as well as lack of missed deadlines. For WebKit in particular, we hope to see a reduction in the amount of time it takes between landing a patch and shipping it out to users, especially since WebKit security affects all browsers used in iOS, as WebKit is the only browser engine permitted on the iOS platform.

For Microsoft, we suspect that the high time to fix and Microsoft's reliance on the grace period are consequences of the monthly cadence of Microsoft's "patch Tuesday" updates, which can make it more difficult for development teams to meet a disclosure deadline. We hope that Microsoft might consider implementing a more frequent patch cadence for security issues, or finding ways to further streamline their internal processes to land and ship code quicker.

Moving forward

This post represents some number-crunching we've done of our own public data, and we hope to continue this going forward. Now that we've established a baseline over the past few years, we plan to continue to publish an annual update to better understand how the trends progress.

To that end, we'd love to have even more insight into the processes and timelines of our vendors. We encourage all vendors to consider publishing aggregate data on their time-to-fix and time-to-patch for externally reported vulnerabilities. Through more transparency, information sharing, and collaboration across the industry, we believe we can learn from each other's best practices, better understand existing difficulties and hopefully make the internet a safer place for all.