Security Now! #858 - 02-15-22
InControl

This week on Security Now!

This week we look at a couple of new 0-days in Chrome and Apple's OSes. We also look at what the US CISA thinks of not only these, but of 15 other problems that our Federal agencies seem to be in no big hurry to fix. And we revisit last summer's Serious SAM vulnerability in Windows which remains under attack. This being the 3rd Tuesday of the month, we'll look back at the 2nd Tuesday to see how that went. Sunday, saw a true emergency patch issued by Adobe that probably canceled some superbowl plans, and we have an amazingly bad idea for a WordPress add-on. Google has published their 2021 bounty report, and their project zero has published stats about how things are going there. We have Microsoft removing a popular and highly abused feature of Windows. And then because nothing else in the past week commanded the podcast's title, I'll wind up by formally introducing GRC's latest freeware which puts its users firmly "InControl".

"I keep writing 'Stone Age' instead of 'Bronze Age' on all my checks."

0-Day Watch

A high-severity 0-day in Chrome

Just yesterday, Google moved Chrome for the Desktop to v98.0.4758.102 to eliminate a high-severity 0-day vulnerability that was spotted being used in attacks. As usual, Google is closed-mouthed about details, saying only that they are aware of reports that an exploit for CVE-2022-0609 — a use after free error in Chrome's support for animation — exists in the wild."

While they were at it, they also fixed seven other security problems, all but one of which were classified as being High severity. And so it goes. Since exploits for today's vulnerabilities are considered highly valuable by those who have them, and since they may have paid a hefty price for it, they're never used in widespread indiscriminate attacks, but only against targeted individuals. Therefore, it's unlikely in the extreme that any of us will be a target. But keeping our browsers updated has become quick, easy and automatic.

So here we are in mid-February with the first of 2022 0-day patched. As we know, lasts year Google addressed 16 0-day vulnerabilities, with the first one patched on Fedruary 4th and the second one on March 2nd. Sso this year is starting out a bit better than last. We'll see how this goes.

Apple updates against another 0-day

Meanwhile, last Thursday Apple released updates for iOS, iPadOS, macOS, and Safari to resolve a flaw in WebKit that was believed to be actively exploited in the wild. So far this year Apple is having a bit rougher time of it than Google with Chrome since this will be the 3rd 0-day it has patched so far. And, boy!... The NSO group sure has been giving Apple a run for their money.

Much like Chrome (and like so many of the flaws we've been encountering recently) the problem in this case is another use-after-free vulnerability granting its attacker arbitrary code execution. And much like Google, Apple is not saying much beyond: “Apple is aware of a report that this issue may have been actively exploited.” They credited an anonymous researcher for discovering and reporting the flaw.

So updates are available for all Apple devices that are receiving updates. As usual, my phone hadn't yet gotten around to installing it. v15.3.1 was waiting. So you might check your various iDevices if you're curious.

And CISA thinks this Apple vulnerability is quite serious...

The US Cybersecurity and Infrastructure Security Agency (CISA) has added that Apple 0-day to its short list catalog of vulnerabilities being exploited in the wild. And they may know something that Apple's not saying. According to the CISA's binding operational directive (BOD 22-01) federal agencies are now required to patch their systems against this actively exploited vulnerability impacting iOS, iPadOS, and macOS devices. CISA said that all Federal Civilian Executive Branch Agencies have until February 25th, 2022 to fully patch against this vulnerability.

CISA said: *"These types of vulnerabilities are a frequent attack vector for malicious cyber actors of all types and pose significant risk to the federal enterprise."* And they added that *"Although BOD 22-01 only applies to FCEB agencies, CISA strongly urges all organizations to reduce their* *exposure to cyberattacks by prioritizing timely remediation of Catalog vulnerabilities as part of their vulnerability management practice."*

What's interesting is that this CISA catalog is a short list. It contains only 16 vulnerabilities for which patches exist, but which are also known to be under active exploitation in the wild.

Which brings us back to "SeriousSAM" as it's being called.
Last Thursday, CISA also asked all Federal Civilian Executive Branch agencies to patch CVE-2021-36934 which is the Microsoft Windows Security Accounts Manager (SAM) bug that allows privilege escalation and credential theft. And for this one they gave a February 24th patch deadline. In other words, "you all have two weeks" from last Thursday.

We talked about SeriousSAM when the news of it broke last summer. As its name suggests, it's serious. It's an elevation of privilege vulnerability which was introduced into all Windows client and server editions released since October 2018, therefore starting with Windows 10 1809 and Windows Server 2019. And the vulnerability was very irresponsibly disclosed, in a tweet. So, despite not being exploited at the time, it would be what Microsoft would designate as a 0-day — since it came as an unwelcome surprise to them when they read about it on Twitter.

The problem arose due to overly permissive Access Control Lists (ACLs) which were present on multiple security-critical system files, including the Security Accounts Manager (SAM) database. An attacker who successfully exploits this vulnerability could have full run of the system, able to run arbitrary code with SYSTEM privileges which, of course, allows such an attacker to install programs; view, change, or delete data; or create new accounts with full user rights.

And what might jog your memory about this one in particular, is that simply correcting the too-permissive access control lists doesn't eliminate it fully. After installing the patch, it's necessary to then manually delete all volume shadow copies of system files, including the SAM database. Since the trouble is with their default permissions, which are stored with the files, if those backup files (with their permissive permissions) can be restored, so, too, can the trouble.

And this all happened when we talked about it last July. Believe it or not, here we are eight months later and the US CISA is finally having to lower the boom on Federal Civilian Executive Branch Agencies, giving them until February 24th, which is a week from next Thursday, to get this resolved once and for all. How can it possibly be that a serious vulnerability which received an emergency patch and generated a great deal of news last July 20th, has not yet been fixed? ANY update to Windows since then would have incorporated that fix. Can it possibly be that there are US Federal Civilian Executive Branch Agencies that have not applied any updates to Windows since last summer?

The CISA Top 16 list
I think that it's worth taking a look at that CISA Top 16 list of things that really must be fixed by those agencies over which it has jurisdiction. As we can see in the table on the next page, we have a mixture of old and new vulnerabilities. Some, oldies but goodies, based upon their CVE dates, are from as far back as 2014. We have one from 2014, three from 2015, one from 2016, seven from 2017, one from 2018, somehow 2019 sneaked by without any, then one from 2020 and that SeriousSAM vulnerability from last year, 2021...

CVE ID	Description	Patch Deadline
CVE-2021-36934	Microsoft Windows SAM Local Privilege Escalation Vulnerability	2/24/2022
CVE-2020-0796	Microsoft SMBv3 Remote Code Execution Vulnerability	8/10/2022
CVE-2018-1000861	Jenkins Stapler Web Framework Deserialization of Untrusted Data	8/10/2022
CVE-2017-9791	Apache Struts 1 Improper Input Validation Vulnerability	8/10/2022
CVE-2017-8464	Microsoft Windows Shell (.lnk) Remote Code Execution	8/10/2022
CVE-2017-10271	Oracle Corporation WebLogic Server Remote Code Execution	8/10/2022
CVE-2017-0263	Microsoft Win32k Privilege Escalation Vulnerability	8/10/2022
CVE-2017-0262	Microsoft Office Remote Code Execution Vulnerability	8/10/2022
CVE-2017-0145	Microsoft SMBv1 Remote Code Execution Vulnerability	8/10/2022
CVE-2017-0144	Microsoft SMBv1 Remote Code Execution Vulnerability	8/10/2022
CVE-2016-3088	Apache ActiveMQ Improper Input Validation Vulnerability	8/10/2022
CVE-2015-2051	D-Link DIR-645 Router Remote Code Execution	8/10/2022
CVE-2015-1635	Microsoft HTTP.sys Remote Code Execution Vulnerability	8/10/2022
CVE-2015-1130	Apple OS X Authentication Bypass Vulnerability	8/10/2022
CVE-2014-4404	Apple OS X Heap-Based Buffer Overflow Vulnerability	8/10/2022

All of our old friends are here. The oldest pair from 2014 & 2015 are Apple OS X vulnerabilities. We have that HTTP.SYS RCE from 2015 as well as that D-Link DIR-645 router RCE. 2017 had the SMBv1 RCE's, an RCE in Office and that raft of Win32K privilege escalation vulnerabilities that that Polar Bear hacker girl, remember? "SandboxEscaper" kept annoying Microsoft with. There was also a Windows .LNK shortcut link file remote code execution in 2017, and I recall being stunned that we were still having those. Apache Struts had that improper input validation problem at the end of 2017. And recall that SMBv3 RCE in 2020.

Anyway... all of our old friends are still and what's astonishing is that CISA has them all selected out and listed in their "must fix by a deadline" chart because, believe it or not, this is the list of vulnerabilities that are still being exploited **most** today. CISA says, if these get patched things will get much better. And, in the far right Patch Deadline column we see that aside from the single "must patch by Thursday after next", all of the other 15 share the same deadline date of Wednesday, August 10th, 2022. I have no idea who came up with that date. I suspect that it was some number of months from the date that this commandment was first dropped upon the Federal agencies. But in any event, it would be great to have our government's systems patched. It's amazing how much it takes to get that done.

And speaking of patching...

Patch Tuesday

Last Tuesday was the industry's monthly Patch extravaganza. And Microsoft did not disappoint. They fixed 51 vulnerabilities occurring in Windows, Office, Teams, Azure Data Explorer, Visual Studio Code, and Windows Kernel components. Among the 51 defects resolved, 50 are rated Important and one is rated Moderate. And if you're thinking to yourself “Wait, nothing Critical? No 0-days to rush to fix?” you would be right to be surprised and also correct in that observation. Last week was a rare Patch Tuesday. Microsoft also fixed an additional 19 flaws in their Chromium-based Edge web browser.

None of the security vulnerabilities are known to be actively exploited, though one of the flaws fixed is what Microsoft calls a 0-day inasmuch as it was publicly disclosed even though it wasn't yet exploited. But yeah, it could happen. That one is a privilege escalation bug in the Windows Kernel.

There were also a handful of remote code execution vulnerabilities. One in Microsoft's DNS server had a CVSS of 8.8. SharePoint SharePoint Server also had an RCE rating a CVSS of 8.8. Windows Hyper-V's was a 5.3, and the HEVC Video Extensions had three of its own with CVSS's of 7.8.

Azure Data Explorer contained a spoofing vulnerability with a CVSS of 8.1, Outlook contained a security bypass vulnerability (CVSS of 5.3) as did OneDrive for Android with a CVSS of 5.9. Dot Net and Teams both had denial-of-service vulnerabilities with CVSS's of 7.5.

The world is not yet done with Microsoft's Print Spooler, which saw four elevation of privilege flaws fixed, as well as one in the Win32K driver with a CVSS of 7.8. And that Win32K elevation was tagged as being more likely to be exploited because last month a similar vulnerability in the Win32K driver did come under attack after it was patched. So Microsoft is suggesting that this one is likely to be, too.

I mentioned at the top that Patch Tuesday has evolved over the years to become an industry-wide event. Besides Microsoft, security updates were also released by, in alphabetical order, Adobe, Android, Cisco, Citrix, Intel, the various Linux distributions from Oracle, Red Hat, and SUSE, Mozilla's Firefox, SAP, Schneider Electric and Siemens.

And ya gotta wonder how IT folks get anything gone on these 2nd Tuesdays and the days immediately following. Still, it's definitely better than having randomly released updates occurring throughout the month at unpredictable times and with unpredictable urgency.

Security News

The Magneto Emergency

I would imagine that a bunch of people had their Superbowl Sunday plans ruined, some because the buck stopped with them for the creation and release of an emergency, five-alarm CVSS 9.8 patch of a 0-day remote code execution bug in the Magneto 2 / Adobe eCommerce platforms which was being actively exploited in the wild. And on the receiving end, the screaming need to immediately apply the emergency patch to their eCommerce sites would have been important enough to kill their football watching plans.

To set the stage a bit, since we've never talked about the Magneto eCommerce platform, it's open-source and written in PHP. It employs multiple other PHP frameworks such as Laminas and Symfony. The Magento source code is distributed under Open Software License (OSL) v3.0. And, after bouncing around and changing hands a few times, Magento was most recently acquired four years ago by Adobe, back in May of 2018 for a cool $1.68 billion. Not a bad price for some volunteer-developed open source software that can be freely downloaded. Wow.

And Magneto's stats are impressive. As of 2019, more than 100,000 online stores had been created using this platform. The platform code has been downloaded more than 2.5 million times, and $155 billion worth of goods have been sold through Magento-based systems during the year 2019 alone. Magento accounts for approximately one third of all eCommerce.

SanSec, which focuses upon eCommerce security, titled yesterday's Monday announcement: *"Magento 2 critical vulnerability (CVE-2022-24086)"* and they explained: *"This Sunday (Feb 13th) Adobe released an emergency patch for a critical vulnerability in Magento 2. It allows unauthenticated remote code execution (RCE), which is the worst possible type. Actual abuse has already been reported. Adobe has been aware of the issue since at least January 27th [two and a half weeks ago] but decided to issue a patch on Sunday, which is highly unusual. Sansec expects that mass scanning and exploitation will happen within the next 72 hours."*

Under *"Implications"* they wrote: *"This vulnerability has a similar severity as the Magento Shoplift vulnerability from 2015. At that time, nearly all unpatched Magento stores globally were compromised in the days after the exploit publication."*

The trouble is, the fix is a few lines of PHP. So it's not going to take a high-end reverse-engineering hacker to figure out what got patched. And since the cat's already out of the bag, SanSec saw no need **not** to show the full patch in their own announcement. It's just a regular expression string replacement, presumably to sanitize some dangerous user-controlled input.

The patching advice itself is sort of interesting. First of all, the trouble affects versions version 2.3.7-p2 and earlier and 2.4.3-p1 and earlier of both the Magneto and Adobe eCommerce platforms, according to the advisory. And according to SanSec who clearly understand what's going on...

- Sites running Magento 2.3 or 2.4 should install the custom patch from Adobe ASAP, ideally within the next few hours.

- Sites running a version of Magento 2 between 2.3.3 and 2.3.7 should be able to manually apply the patch, as it only concerns a few lines of PHP source.

- And sites running Magento 2.3.3 or below are not vulnerable. However, SanSec still recommends manually implementing the given patch — though you can probably safely watch the big game first.

Online eCommerce sites and vulnerabilities are particularly high-value targets with credit-card skimming being the primary goal. The most active nefarious group in this space is known as the Magecart group which specializes in targeting unpatched versions of Magento in particular, looking for a way to plant credit-card skimmers on the checkout pages of eCommerce websites.

This Magecart group, which is actually a consortium of many different card-harvesting subgroups, consistently evolves its skimmers to be more effective and efficient at evasion. For example, in November, it added an extra browser process that uses the WebGL JavaScript API to check a user's machine to ensure it's not running on a virtual machine to evade researcher detection. And last month, in January, an attack on Segway planted a skimmer by using a favicon that traditional security systems wouldn't inspect.
Adobe, who is working to downplay the apparent sudden severity of the issue after having sat in for more than two weeks has initially characterized the attacks as "very limited." But card-skimmer activity is on the rise, and we know how long web sites often take to update. And for example, completely separate from this, last week SanSec reported a wave of skimming attacks targeting more than 500 sites, in particular those using outdated and unsupported Magento 1 implementations and data from Source Defense found as many as 100,000 sites that are still using past end-of-life Magento v1. Ouch.

"PHP Everywhere"
While we're on the subject of things written in PHP, yet another WordPress add-on, this one called "PHP Everywhere" — which really doesn't sound like a good idea — was responsible for placing more than 30 thousand additional WordPress sites at risk of remote code execution.

The fact that Magneto, which we were just talking about, powers about 1/3rd of the world's highly targeted eCommerce sites, was also written in PHP, demonstrates that it is definitely possible for sufficiently skilled developers to author secure website code in PHP. That's not the issue. The issue is that PHP's deliberately and seductively easy-to-use design, which is what makes it so popular, encourages unskilled developers to place their code online. That pithy little slogan I coined a few weeks ago comes to mind: *"Most developers stop working the moment their code starts working."* We know that there's a big gap between code which works and code which is also secure against attack. And due to the online website environment where PHP typically finds itself, that often leads to trouble. I now have a bunch of stuff at GRC written by other people in PHP. And the only way I was going to ever allow any of that near GRC's network was by putting it all on its own physical server located behind a physical firewall that almost completely cuts it off from the rest of GRC's network. If something gets loose in there at least I have containment.

So today we have "PHP Everywhere." Get a load of this. This is like "Meta Bad": The WorldPress plugin PHP Everywhere, is named as it is, because it deliberately allows site owners to execute PHP code anywhere on their site. According to the add-on's description:

"This plugin enables PHP code everywhere in your WordPress installation. Using this plugin you can use PHP in: Pages, Posts [In POSTS! You can put PHP in Posts! What could possibly go wrong?]*, and also in the Sidebar.?*

Its description says "Everywhere." And it goes on to explain:

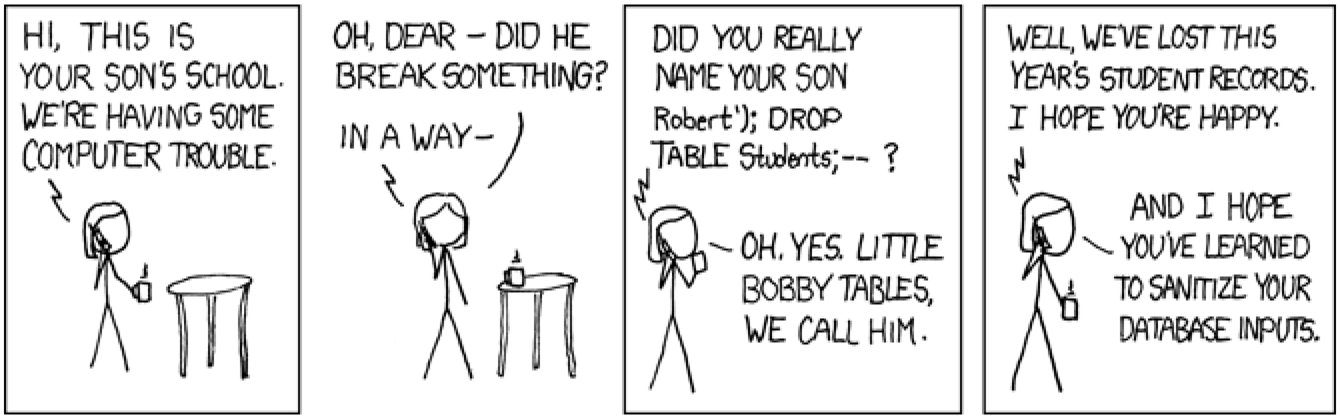
The plugin also supports different user restrictions and multiple PHP instances. So feel free to just insert PHP in every part of your WordPress site. Examples of use:

- *Create custom contact forms and process any kind of data or upload.*
- *Generate user optimized content.*
- *Customize every little detail of your WordPress installation.*

That's right, create a handy dandy WordPress add-on that encourages site operators, who may barely be able to code, to liberally litter PHP everywhere around their site, and you, too, can earn not just one, but three of those rare CVSS scores of 9.9 accompanied by three of your own CVE's: CVE-2022-24663, 664 and 665.

And, as it turned out, PHP Everywhere's functionality allowed the execution of PHP Code Snippets through WordPress shortcodes. Unfortunately, WordPress allows any authenticated users to execute shortcodes via the parse-media-shortcode AJAX action, and some plugins also allow unauthenticated shortcode execution. As such, it was possible for any logged-in user, even a user with no permissions, such as a Subscriber or a Customer, to execute arbitrary PHP on a site by sending a request with the shortcode parameter set to [php_everywhere] (to invoke the add-on) then <arbitrary PHP> then [/php_everywhere]. And, not surprisingly, executing arbitrary PHP on a site typically allows complete site takeover.

Allowing users to execute their own code is reminiscent of all of the persistent problems we used to have with SQL injection. That inspired this classic XKCD cartoon:

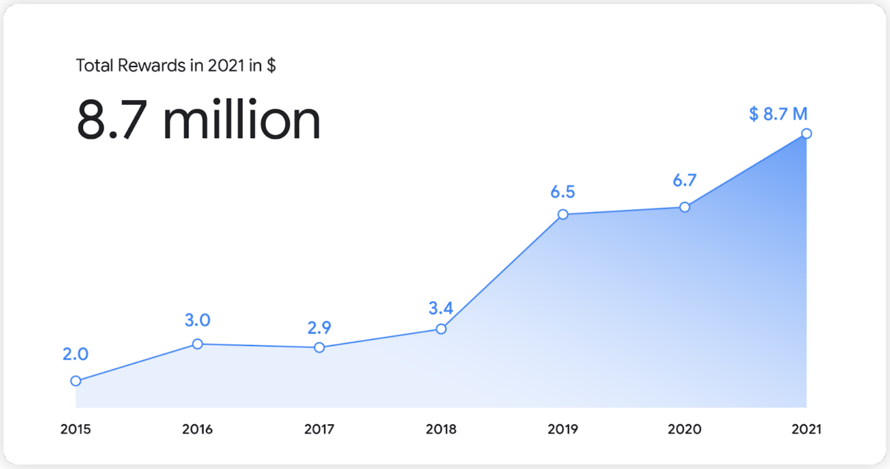

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

https://xkcd.com/327/

So this is a perfect case in point. This should have never been allowed to happen. But there's no oversight, and the deliberately created ecosystem surrounding WordPress actively encourages this sort of thing. Since none of my ranting is likely to change WordPress's approach one iota, my only hope — and my purpose here — is to adequately instill in our listeners a sober appreciation for the dangers inherent in using 3rd-party add-ons with WordPress.

It's clear that the base WordPress system itself is mature, was professionally written and is being professionally maintained. It's secure and highly bulletproof. But that security doesn't necessarily pertain, in any way, to anything that's added to it.

Google's Vulnerability Reward Program for 2021
Last Thursday, Google share the results of their vulnerability responsible reporting reward program for 2021:
https://security.googleblog.com/2022/02/vulnerability-reward-program-2021-year.html



There are a bunch of interesting facts and stats (and dollar amounts and bug counts) that I believe our listeners will find interesting. So I'm going to share an edited-down version of Google's posting:

Last year was another record setter for our Vulnerability Reward Programs (VRPs). Throughout 2021, we partnered with the security researcher community to identify and fix thousands of vulnerabilities – helping keep our users and the internet safe.
Thanks to these incredible researchers, Vulnerability Reward Programs across Google continued to grow, and we are excited to report that in 2021 we awarded a record breaking $8,700,000 in vulnerability rewards – with researchers donating over $300,000 of their rewards to a charity of their choice.

We also launched https://bughunters.google.com/ in 2021, a public researcher portal dedicated to keeping Google products and the internet safe and secure. This new platform brings all of our VRPs (Google, Android, Abuse, Chrome, and Google Play) closer together and provides a single intake form, making security bug submission easier than ever. We're excited about everything the new Bug Hunters portal has to offer.

Android
The Android VRP doubled its 2020 total payouts in 2021 with nearly $3 million dollars in rewards, and awarded the highest payout in Android VRP history: an exploit chain discovered in Android receiving a reward of $157,000!

Our industry leading prize of $1,500,000 for a compromise of our Titan-M Security chip used in our Pixel device remains unclaimed - for more information on this reward and Android exploit chain rewards, please visit our public rules page.

The program also launched the Android Chipset Security Reward Program (ACSRP), a vulnerability reward program offered by Google in collaboration with manufacturers of certain *popular Android chipsets. This private, invite-only program, provides reward and recognition for contributions of security researchers who invest their time and effort into helping make Android devices more secure. In 2021 the ACSRP paid out $296,000 for over 220 valid and unique security reports.*

We would like to give a special shoutout to some of our top researchers whose continued hard work keeps Android safe and secure:

- *Aman Pandey of the Bugsmirror Team has skyrocketed to our top researcher last year, submitting 232 vulnerabilities in 2021! Since submitting their first report in 2019, Aman has reported over 280 valid vulnerabilities to the Android VRP and has been a crucial part of making our program so successful.*

- *Yu-Cheng Lin (林禹成) (@AndroBugs) has been another phenomenal researcher for the Android VRP, submitting a whopping 128 valid reports to the program in 2021.*

- *Researcher gzobqq@gmail.com discovered a critical exploit chain in Android (CVE-2021-39698), receiving the highest payout in Android VRP history of $157,000.*

Chrome
This year the Chrome VRP also set some new records – 115 Chrome VRP researchers were rewarded for 333 unique Chrome security bug reports submitted in 2021, totaling $3.3 million in VRP rewards. The contributions not only help us to improve Chrome, but also the web at large by bolstering the security of all browsers based on Chromium.

Of the $3.3 million, $3.1 million was awarded for Chrome Browser security bugs and $250,500 for Chrome OS bugs, including a $45,000 top reward amount for an individual Chrome OS security bug report and $27,000 for an individual Chrome Browser security bug report.

Of these totals, $58,000 was awarded for security issues discovered by fuzzers contributed by VRP researchers to the Chrome Fuzzing program. Each valid report from an externally provided fuzzer received a $1,000 patch bonus, with one fuzzer report receiving a $16,000 reward.

The Chrome VRP would not be able to smash these records over the last year without the efforts of so many exceptional VRP researchers. We'd like to highlight a few researcher achievements made in 2021:

- *Rory McNamara, a Chrome OS VRP researcher who has been participating in the Chrome VRP for five years, became the highest awarded Chrome VRP researcher of all time. This year he was rewarded for six reports achieving root privilege escalation in Chrome OS, one of which received the highest reward amount achieved for a single Chrome bug report in 2021 at $45,000.*

- *Chrome Browser VRP researcher Leecraso (@leecraso) of 360 Vulnerability Research Institute was the most awarded researcher of 2021, with 18 valid bug reports; a majority of which were for memory corruption vulnerabilities affecting the browser process.*

- *We love when researchers write about their findings (only after we have publicly disclosed the bug, of course)! Chrome Browser VRP researcher Brendon Tiszka wrote an excellent two-part blog series on his discovery and exploitation of a V8 vulnerability, CVE-2021-21225, the analysis and reporting of which earned him a $22,000 VRP reward.*

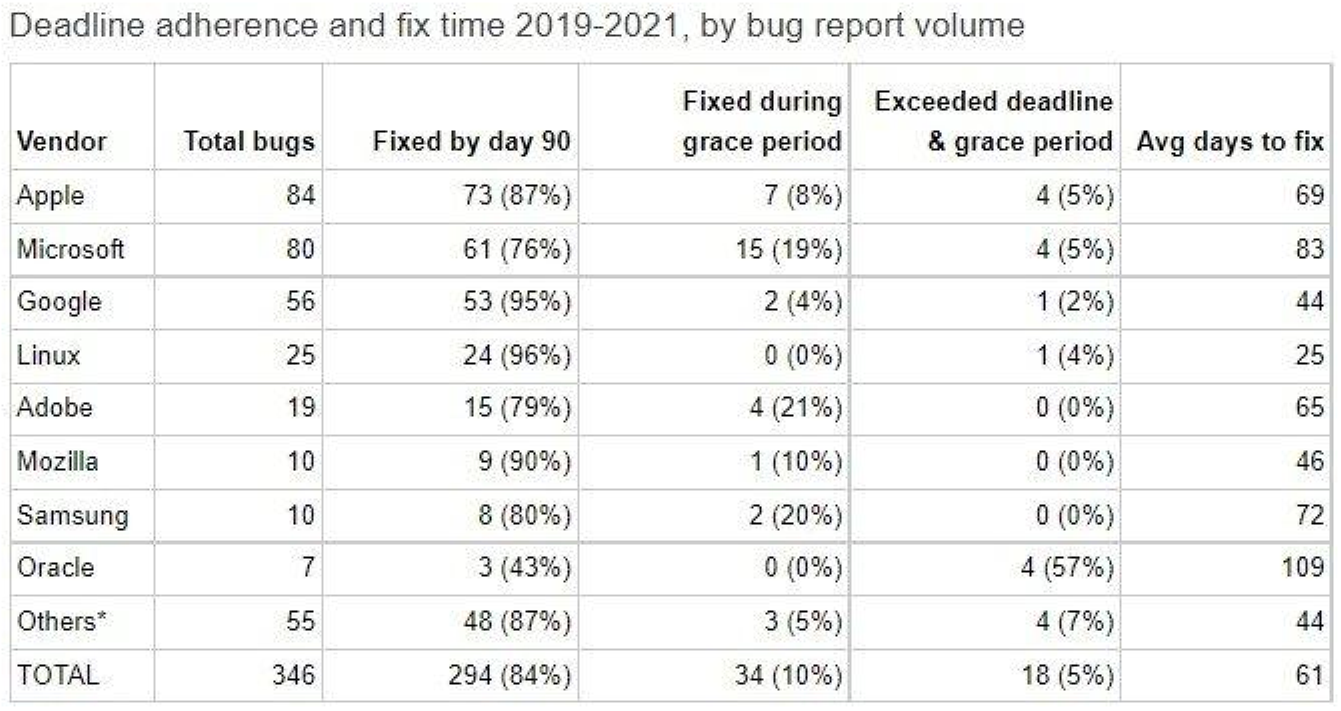
Huge thanks and congratulations to all Chrome VRP researchers that helped us make Chrome and Chrome OS more safe for all users in 2021!.

So... $8.7 million last year. That's some serious coin. Bug hunting is not guaranteed income, but it might be an interesting way to spend some spare evenings when nothing else is going on. The more you look and poke around the more you'll learn. And as I've said before, there's really no better way to extend one's own coding skills than by reading and comprehending someone else's code. And, hey ... if you find something wrong, you might have a well-earned payday!

Google's Project Zero Stats
While we're talking stats from Google, it's worth noting that Google has reported that vendors are getting quicker about fixing 0-day flaws in their offerings. And there were some other interesting stats...

Google's recently published data reveals that the average period software vendors used to repair and issue security updates reported by Project Zero last year was 52 days, which was a significant reduction — despite no reduction in bug levels — from 80 days three years ago. And now, nearly all vendors are addressing their flaws within what has become an accepted industry deadline of 90 days, plus a grace period of two weeks.

The stats for the 2019-2021 period show a total of 376 0-day reports from Project Zero with 26% concerning Microsoft, 23% Apple, and 16% Google. The sum of those three is 65% of all project zero events — Microsoft, Apple and Google — the largest commercial desktop OS vendor and the two biggest mobile OS vendors.


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

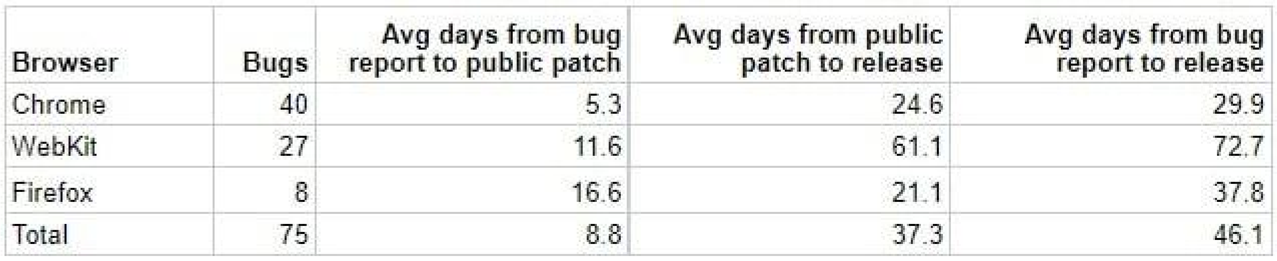
Zero-day fixing stats from 2019-2021 *(Google)*

I have a chart in the show notes which breaks the stats down by vendor. Oracle had the fewest project zero reported 0-days at only 7, though they also took the longest to fix those with four of them exceeding both the 90-day fix-by deadline and the additional 2-week grace period.

As the chart above shows, the three overall best patching performers by average days to fix were Linux at just 25 days, Google at 44 day, and Mozilla at 46. At the worst performing end was Oracle at 109 days (but with only those 7 problems compared to Apple who had 84 0-days) then Microsoft who took an average of 83 days each and Samsung who was a bit quicker at 73 days. And, notably, Microsoft also had the most fixes (15 of them) occurring within that final 2-week grace period, so just squeaking under the full disclosure deadline.

And comparing the mobile OS terrain, iOS and Android were about tied, with iOS having an average time-to-fix of 70 days and Android taking 72 days.

As for web browsers, not surprisingly, since we often observe Google's quick response with Chrome, they beat everyone with an average time-to-fix of 29.9 days and on the other end is Apple's Webkit at 72.7 days average. At the same time, that was 40 bugs for Chrome and only 27 for Webkit.



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

But, again, as an industry overall, it seems clear that despite the appearance of an increasing rate of problems — and it feels like an increase in problem severity since we're dragging an ever growing legacy behind us — those tasked with fixing problems, much as we might wish they were even better, are managing to stay ahead, keep things under control, and reduce our overall exposure.

Bye bye WMIC (and once again, yes, "WMIC-key, mouse")
Last week we talked about the practice of "Living off the Land", where malware or miscreants take advantage of commands and features available at the local system. One of the more often abused features of Windows, which we touched on last week, is the massively capable Windows Management Instrumentation (WMI) system which is accessible through the Command-line executable: wmic.exe.

The fact that this useful command-line access tool is so often abused has not escaped Microsoft's attention. And as a result, wmic.exe is going away. And if you know anything about Microsoft, it's that they really really really really really really strongly dislike ever removing anything from Windows, especially something like wmic, a system management tool that almost certainly figures actively in the management scripts being used by their enterprise users, among others.

Despite this extreme reluctance, Microsoft is in the process of removing wmic.exe, starting with the latest Windows 11 preview builds in the Dev channel. They did announce last year that this was their intention, and that they were in the process of deprecating the use of wmic.exe in Windows Server in favor of Windows PowerShell, which provides a full superset of WMIC's capabilities including the ability to query the Windows Management Instrumentation system itself.

Microsoft wrote: *"The WMIC tool is deprecated in Windows 10, version 21H1 and the 21H1 General Availability Channel release of Windows Server. This tool is superseded by Windows PowerShell for WMI. This deprecation only applies to the command-line management tool. WMI itself is not affected."*

So now, Microsoft has been spotted removing WMIC from Windows clients, starting with Windows 11 preview builds in the Dev channel. The guys at BleepingComputer did some sleuthing and confirmed that from at least build 22523, the WMIC command is no longer available in 'Dev' channel clients, though they noted that Microsoft may have removed it from earlier builds.

Microsoft may be testing the waters to see whether they're able to pull it from Windows, or to gauge how much fur will fly if they do. They clearly want to, and you'll note that the only possible reason for deliberately yanking something from Windows that's already there and that many good people are using, is that many bad people are using it, too.

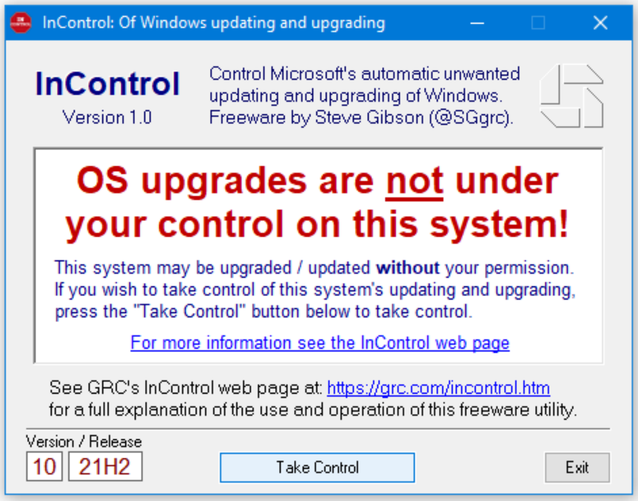
One thing that might happen, which wouldn't surprise me, would be for them to make it available separately as an optional manual download. That's something they've done before. It's certainly the case that 99.99% of Windows client users have never typed the command "wmic" and never will. So having it there, only to ever be used by malware, really makes no sense. But leaving it as optionally available, though deprecated and not recommended, minimizes the damage from pulling it entirely.

In any event, for maximum compatibility, everyone should take this as a heads-up. If you or your enterprise are currently dependent upon WMIC, you'd be well advised to move to PowerShell.

InControl

https://www.grc.com/incontrol.htm

Introducing "InControl"



InControl: Of Windows updating and upgrading

InControl Version 1.0

Control Microsoft's automatic unwanted updating and upgrading of Windows. Freeware by Steve Gibson (@SGgrc).

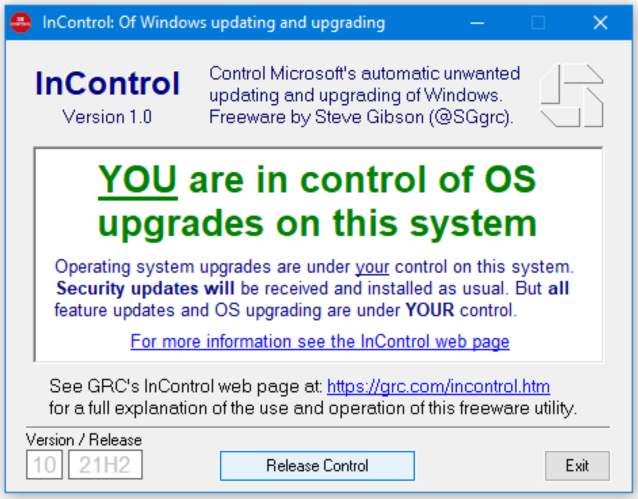
OS upgrades are not under your control on this system!

This system may be upgraded / updated **without** your permission. If you wish to take control of this system's updating and upgrading, press the "Take Control" button below to take control.

For more information see the InControl web page

See GRC's InControl web page at: https://grc.com/incontrol.htm for a full explanation of the use and operation of this freeware utility.

Version / Release 10 21H2



InControl: Of Windows updating and upgrading

InControl Version 1.0

Control Microsoft's automatic unwanted updating and upgrading of Windows. Freeware by Steve Gibson (@SGgrc).

YOU are in control of OS upgrades on this system

Operating system upgrades are under your control on this system. **Security updates will** be received and installed as usual. But **all** feature updates and OS upgrading are under **YOUR** control.

For more information see the InControl web page

See GRC's InControl web page at: https://grc.com/incontrol.htm for a full explanation of the use and operation of this freeware utility.

Version / Release 10 21H2

- Never10 —> Never11 —> StayPut —> UnderControl —> TakeControl —> InControl.
- 82kb of x86 assembly code.
- No setup or install.
- Manages six different Registry key values to get Windows Update under control.