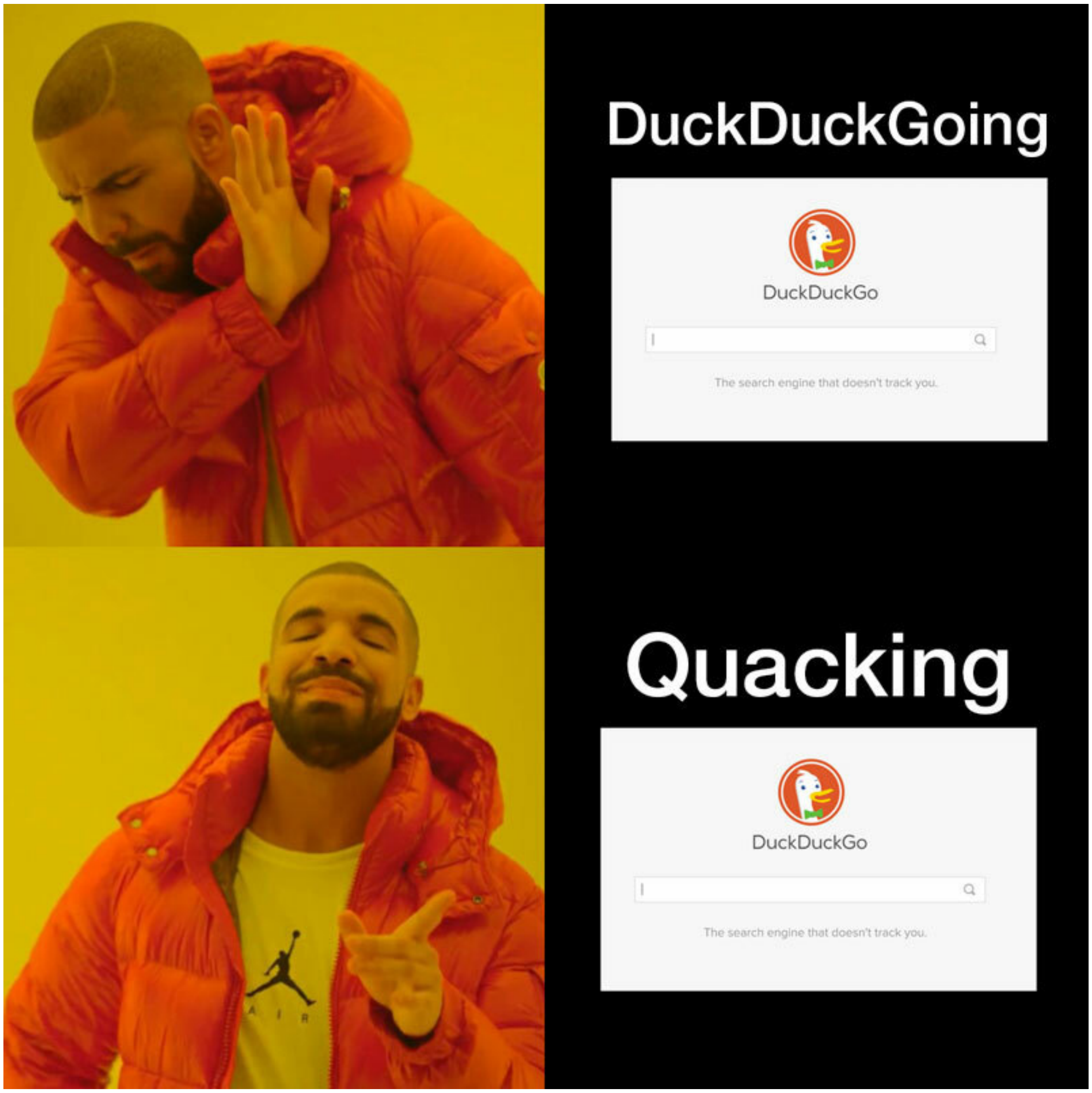
Security Now! #868 - 04-26-22

The 0-Day Explosion

This week on Security Now!

This week we're going to take a close look at the U.S. Cybersecurity and Infrastructure Security Agency's mandated must update list, including some recent entries. We're going to examine the somewhat breathtaking mistake that Lenovo made across more than 100 of their laptop models, and a cryptocurrency wallet implemented in a web browser (what could possibly go wrong?) Then we're going to look at another startling vulnerability that was recently discovered in Java versions 15, 16, 17 and 18. We have a bunch of interesting listener feedback, a brief Sci-Fi interlude, and the announcement of a major milestone reached for SpinRite. Then we're going to wrap up by taking a look across the past ten years of 0-day vulnerabilities thanks to some recent research performed by the security firm Mandiant. The title of this week's podcast gives away what's been happening.

"Ducking?"


DuckDuckGoing
DuckDuckGo
The search engine that doesn't track you.
Quacking
DuckDuckGo
The search engine that doesn't track you.
AIR

Security News

CISA's Known Exploited Vulnerabilities Catalog

As we've noted from time to time, one of the services being provided by our awkwardly named U.S. Cybersecurity and Infrastructure Security Agency (CISA) has been the maintenance of a growing list of actively exploited vulnerabilities. Again, this is not just a list of vulnerabilities, but specifically a list of vulnerabilities that are being seen exploited in the wild. We normally refer to this list in the context of CISA's "Christmas Canceling" policy of issuing standing mandates to all federal agencies within its governance that they =MUST= patch this or that by a specific date, typically only a few weeks after the issuance of the mandate.

At the end of today's podcast we're going to be talking a bit about patching philosophy and about the idea of prioritizing patching to focus upon those issues that are being actively exploited. The logic behind that is obvious and in a bureaucratic environment it certainly makes the imposition of the inconvenience that's suffered by the need to take down and patch running systems and services more justifiable.

But boy, CISA's list is growing so large that it's now being referred to as a "catalog." So at some point it loses some of its punch as it becomes easier just to patch everything, which, as we'll see, is the strategy that I think makes the most sense, overall.

That said, there have been some notable new entries added to CISA's constantly growing catalog of "must patch immediately" mandates:

CISA informs us that the CVSS 7.8 vulnerability in Windows Print Spooler that was patched as part of February's Patch Tuesday — so more than 60 days ago, we've had March and April patches since then — is currently being actively exploited in the wild. It's a privilege escalation vulnerability which, as we know, a hacker will need to leverage if they're able to arrange to get into a system but under limited protective privileges.

So here's a perfect example of a more than 60-day old patched problem that was doubtless immediately reverse engineered and put to use. The only place it's going to be effective is against machines that have not yet received their February updates... yet being exploited in the wild, it is (says Yoda.)

Back in February when Microsoft listed this defect as fixed it was tagged as "exploitation more likely" and they were clearly right about that.

It's interesting, and somewhat sad, to look at the CVSS year dates for things the CISA adds to its actively being exploited in the wild catalog. For example, they just added a cross site scripting vulnerability which was found in the "Zimbra Collaboration Suite." It's certainly not mainstream. It's Java-based and Linux hosted. It's been around since 2005 and its ownership has changed hands many times. It was first written by a company originally named LiquidSys who chanced their name to Zimbra, which Yahoo! later purchased, before selling it to VMWare who sold it to Telligent Systems, who then also changed their name to Zimbra before being sold to Synacor. I guess lots of companies had great hopes for it. But in any event, this cross-site scripting vulnerability was identified and patched four years ago in 2018 and CISA tells us that it is currently being actively exploited in the wild.

And guess where? Ukraine's Computer Emergency Response Team (CERT-UA) released an advisory last week cautioning about eMail phishing attacks targeting government entities with the goal of forwarding victims' emails to a third-party email address by leveraging exactly that Zimbra vulnerability. So, okay, it's not just theoretical. As CISA said, it's actually happening right now. Some Russian miscreant saw that someone was still using Zimbra. So they checked their catalog titled: “The Big Book of Every Possible Way to Hack Someone” and found a 4-year old XSS scripting vulnerability that would come in handy if that Zimbra instance had not been updated in the past four years. The most recent Zimbra update was just a little over one year ago, on April 7th, 2021. So Zimbra could have been kept current. But apparently it hadn't been.

So, CISA's not wrong that this 4-year old obscure vulnerability is being exploited, but are they right to add it to the U.S.'s emergency “must patch by May” mandate? Are any U.S. Federal agencies known to be using Zimbra? I have no idea. But if so, wouldn't it make much more sense to be a bit proactive and to target those agencies rather than everyone else with an entry that just adds unnecessary noise to the growing list?

Another just-added entry, is a 3-year old stack buffer overflow carrying a CVSS of 9.8. 9.8!! That's a stand up and take notice score, and it occurs in WhatsApp's VOIP component. How is it possible that anyone using WhatsApp will not have updated it since 2019?

Leo, I guess that perhaps we're living in a bubble. Perhaps we've been drinking our own “Upgrade CoolAid” for so long that we're completely out of touch with the real world. There must be a significant proportion of users who actively and proactively ignore, or perhaps mistrust, offers and requests to update their software. Maybe if it's working and not obviously broken they see no need to mess with it?

As we know, it certainly is the case that having something work and having something that works also being actively impervious to abuse are two very different things. But WE know that. That's one of the most important lessons that everyone listening to this podcast — myself included — has learned through example after example through the years. But that's probably not at all obvious to the typical user who thinks “Well, it works. I can message and talk and it seems fine here, so whatever they're trying to sell or do I probably don't need or want. So I'm not going to change anything.”

But even so, does a 3-year old vulnerability, even if someone in Bangladesh had their Android phone compromised as a result of using an even older version of WhatsApp, really deserve to be taking up cognitive space in CISA's catalog? I think it's an open question.

Three or four years is a LONG time to not have updated. It's clear that there's a very long tail on many of these vulnerabilities. We talked last year and the year before about critical flaws in a TCP/IP stack being widely used by $5 IoT light switches and plugs. None of those things are ever going to be updated. So they will be latently vulnerable as long as they're in service. But that isn't what CISA is targeting.

I don't have an answer. Light switches and plugs cannot be updated. But for Federal agencies to never update in-use software for which updates are ready and waiting is unconscionable. I'm glad there's a mandate. I hope it has some teeth behind it.

Lenovo UEFI Firmware Troubles

When a PC is powered up, something needs to wake up and configure the various parts of the machine. The video needs to be started, the fans need to spin up, all of the machines various mass storage subsystems need to be initialized and then the firmware's configuration needs to be checked, the proper operating system needs to be located and its OS boot code needs to be loaded into RAM and then control turned over to it to take over the machine.

The first PCs did that using their Basic Input Output System, B.I.O.S, or BIOS. That was good for about five years, until the limitations that had been built into the BIOS's assumptions began to cause more problems than they were worth. Various Mickey Mouse workaround were created to overcome many of these problems while Intel worked on a wholesale replacement of the BIOS. The initial attempt was the EFI — Extensible Firmware Interface — which quickly matured into the Unified Extensible Firmware Interface, or UEFI.

And we find ourselves right back where we always are: The original BIOS was so dumb that it could not be infected. It was originally implemented in masked ROM, meaning that the firmware's bits were etched into a metal mask at the factory and could never be changed. That soon gave way to non-volatile FLASH ROM which could be updated, but the code it implemented was still extremely dumb. Sometimes, for some things, the dumber the better. Because if all you want is to boot an OS, you don't really need much smarts. And the lesson we keep failing to learn is that the more complicated, fancy, capable and "smart" we make things, the more leeway and latitude that system has to go very badly wrong.

So, welcome to the Unified Extensible Firmware Interface where malware is also able to "extend" the firmware.

Lenovo has been most recently in the "we made a UEFI mistake" news recently. Last week, the guys over at ESET, whose motto is "We Live Security" posted the results of their analysis of some widely used Lenovo UEFI firmware. Their posting's title was "When "secure" isn't secure at all: High-impact UEFI vulnerabilities discovered in Lenovo consumer laptops." and the story's tag line is: "ESET researchers discover multiple vulnerabilities in various Lenovo laptop models that allow an attacker with admin privileges to expose the user to firmware-level malware."

Firmware Level Malware. That's not what you want to hear. That's even less what you want to have crawling around inside your machine. Firmware level malware enables the ultimate in rootkit techniques, in fact having its own worse name: "Bootkit." The presence of firmware level malware means, quite simply, that it's impossible to trust anything about what the machine might do. Firmware level malware is able to infect and compromise the operating system's own code during its boot process before it has had any opportunity to raise its own shields. And reformatting the machine's mass storage and reinstalling an OS, or even removing and replacing a drive won't eliminate the problem because this malware has taken up residence in the machine's underlying firmware.

Now, we know that anyone can make a mistake. But the most troubling aspect of what the ESET researchers found was that two of the three big mistakes Lenovo made were the oversight of leaving highly exploitable drivers in the UEFI firmware image which should only have been present during firmware development. Those drivers should have never left the factory.

The two drivers were actually named: “SecureBackDoor” and “SecureBackDoorPeim”. Here's what ESET says:

> *ESET researchers have discovered and analyzed three vulnerabilities affecting various Lenovo consumer laptop models. The first two of these vulnerabilities – CVE-2021-3971, CVE-2021-3972 – affect UEFI firmware drivers originally meant to be used only during the manufacturing process of Lenovo consumer notebooks. Unfortunately, they were mistakenly included also in the production firmware images without being properly deactivated. These affected firmware drivers can be activated by an attacker to directly disable SPI flash protections (Control Register bits & Protected Range registers) or the UEFI Secure Boot feature from a privileged user-mode process during OS runtime.*

Just to be clear about what ESET just said: “From a privileged user-mode process in the OS.” In other words, mistakenly allowing some malware to run in the OS, which might innocently ask to be granted brief UAC privilege elevation to install something, or which might set itself up as a system service, can disable all relevant UEFI write protections to surreptitiously install semi-permanent hidden bootkit malware into the system’s UEFI firmware.

> *It means that exploitation of these vulnerabilities would allow attackers to deploy and successfully execute SPI flash or ESP implants, like LoJax.*
>
> *To understand how we were able to find these vulnerabilities, consider the firmware drivers affected by CVE-2021-3971. These drivers immediately caught our attention by their very unfortunate (but surprisingly honest) names: **SecureBackDoor** and **SecureBackDoorPeim**. After some initial analysis, we discovered other Lenovo drivers sharing a few common characteristics with the SecureBackDoor* drivers: ChgBootDxeHook and ChgBootSmm. As it turned out, their functionality was even more interesting and could be abused to disable UEFI Secure Boot (CVE-2021-3972).*
>
> *In addition, while investigating the vulnerable drivers, we discovered a third vulnerability: SMM memory corruption inside the SW SMI handler function (CVE-2021-3970). This vulnerability allows arbitrary read/write from/into SMRAM, which can lead to the execution of malicious code with SMM privileges and potentially lead to the deployment of an SPI flash implant.*
>
> *We reported all discovered vulnerabilities to Lenovo on October 11th, 2021. Altogether, the list of affected devices contains more than **one hundred different consumer laptop models** with millions of users worldwide, from affordable models like Ideapad-3 to more advanced ones like Legion 5 Pro or Yoga Slim 9. The full list of affected models with active development support is published in the Lenovo Advisory.*
>
> *In addition to the models listed in the advisory, several other devices we reported to Lenovo are also affected, but won’t be fixed due to them reaching End Of Development Support (EODS). This includes devices where we spotted reported vulnerabilities for the first time: Ideapad 330 and Ideapad 110. The list of such EODS devices that we have been able to identify will be available in ESET’s vulnerability disclosures repository.*

Lenovo confirmed the vulnerabilities on November 17th, 2021, and assigned them the following CVEs:

- CVE-2021-3970 LenovoVariableSmm – SMM arbitrary read/write
- CVE-2021-3971 SecureBackDoor – disable SPI flash protections
- CVE-2021-3972 ChgBootDxeHook – disable UEFI Secure Boot

Given how incredibly active the cyber underworld is today—we keep encountering quite sobering evidence of it—there's just no chance that these now fully disclosed and very well documented vulnerabilities will not be used to compromise the interests of some of these millions of Lenovo laptop users worldwide. It will happen. So here we are once more noting that there's something very wrong with our industry's current development model. How can this be allowed to occur over and over and over. It's designed to happen.

Lenovo messed up bigtime here, but they're not alone. These newly disclosed vulnerabilities merely add to the recent disclosure of more than 50 UEFI firmware vulnerabilities which have been found in Insyde Software's InsydeH2O, and HP and Dell laptops since the start of the year. Among those are six severe flaws in HP's firmware affecting both laptops and desktops which, when exploited, could allow attackers to locally escalate to SMM privileges and trigger a denial-of-service (DoS) condition. So Lenovo is in good company... or at least only the most recent member of this UEFI vulnerability dog house. And, as we know, it's not Lenovo's first instance of UEFI problems.

So we've made our lovely little machine's far more complex by designing-in extremely powerful capabilities to add flexibility, remote management and maintenance. And guess what... it's a mixed blessing.

So, just a head up to anyone using Lenovo laptops. Regardless of the model you have, you should definitely check-in to see whether your device has a firmware update outstanding. And for that matter, HP and Dell users would be well advised to do the same.

Everscale Blockchain Wallet

I read the title of this piece of news in The Record and it just made me shake my head. The item is titled: "Everscale blockchain wallet shutters **web version** after vulnerability found."

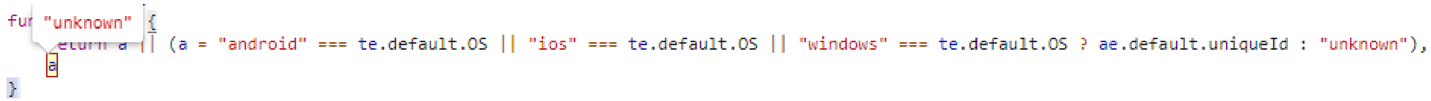
Yeah, no kidding. What moron could possibly think that offering a web browser based cryptocurrency wallet was sane? Anyone who was capable of beginning to create such a thing should know that it's just a bad idea. As we've often observed on this podcast, just because you **can** do something doesn't mean that you **should** do something. Here are the first two sentences of The Record's story:

> *"The company behind Ever Surf, a wallet for the Everscale blockchain ecosystem, is shuttering its web version after a vulnerability was found by Check Point researchers. The Ever Surf team confirmed that the vulnerability allowed attackers to gain access to wallets."*

The Record is reporting on research which was performed by CheckPoint Research. The CheckPoint guys explained:

> *Blockchain technology and decentralized applications provide users with a number of advantages. For example, users can utilize the service without creating an account and it can be implemented as a single-page application written in JavaScript. This type of application does not require communication with a centralized infrastructure, such as a web server, and it can interact with the blockchain directly or by using a browser extension like Metamask.*
>
> *In this case, the user is identified using keys that are stored only on a local machine inside a browser extension or a web wallet. If a decentralized application or a wallet stores sensitive data locally, it must ensure this data is reliably protected. In most cases, decentralized applications run inside the browser and therefore may be vulnerable to attacks such as XSS.*
>
> *This research describes the vulnerability found in the web version of Ever Surf, a wallet for the Everscale blockchain. By exploiting the vulnerability, it's possible to decrypt the private keys and seed phrases that are stored in the browser's local storage. In other words, attackers could gain full control over victims' wallets.*

It turns out that one of the code libraries the implementers used is not fully supported in web browsers. The code attempts to obtain a cryptographic nonce with a call to "DeviceInfo.getUniqueId". The problem is that this function requires access to its underlying device, so it's only defined when running natively on Android, iOS or Windows. I have a snippet of the function in the show notes showing what this one-line function does:



```
fu "unknown" {
    eturn a || (a = "android" === te.default.OS || "ios" === te.default.OS || "windows" === te.default.OS ? ae.default.uniqueId : "unknown"),
    a
}
```

When the OS is not Android, iOS or Windows, the function return the JavaScript pseudo-value "unknown" ... and thus, that value is never unique and that value is used to salt the hash. As we have learned on this podcast eons ago, salting hashes is crucial to the security of hashed passwords because the salt effectively customizes the hash per use. With the salt broken, CheckPoint was able to trivially brute force the user's 6-digit PIN. Yes, on top of everything else, even if the system was working correctly, its entire security was controlled by a 6-digit PIN.

CheckPoint wrote:

> *CPR roughly re-implemented the key derivation and keystore decryption in NodeJS and performed a brute-force attack on the PIN code.*
>
> *This resulted in a performance of 95 passwords per second on 4-core Intel Core i7 CPU. Although this is not a very high speed, it is sufficient for the attack on a 6-digit PIN code. In the worst case scenario, checking 10^6 possible variants means the entire attack takes approximately 175 minutes.*
>
> *For our experiment, we created a new key in Surf and dumped the keystore from the browser's [unencrypted] local storage. In our case, the attack took 38 minutes. At the end, we got the derived key and decrypted the seed phrase that can be used to restore the keys on another device.*

Java 15, 16, 17, and 18 received MUST UPDATES last week.
Neil Madden, the somewhat excitable guy at ForgeRock, who discovered a new and quite severe problem with Java considers it to warrant a CVSS of 10.0. I think we should reserve that for the software apocalypse — or perhaps when SkyNet obtains self awareness. The rest of the industry gave his discovery a very healthy CVSS of 7.5 ... and this one should not be ignored.

Here's what Neil wrote about his discovery:

> It turns out that recent releases of Java were vulnerable to a flaw in the implementation of widely-used ECDSA (Elliptic Curve Digital Signature Algorithm). If you are running one of the vulnerable versions, then an attacker can easily forge some types of SSL certificates and handshakes (allowing interception and modification of communications), signed JWTs (JSON Web Token), SAML assertions or OIDC ID tokens, and even WebAuthn authentication messages. All using the digital equivalent of a blank piece of paper.
>
> It's hard to overstate the severity of this bug. If you are using ECDSA signatures for any of these security mechanisms (and ECDSA is now the default standard, then an attacker can trivially and completely bypass them if your server is running any Java 15, 16, 17, or 18 version before the April 2022's Critical Patch Update (CPU). For context, almost all WebAuthn/ FIDO devices in the real world (including Yubikeys*) use ECDSA signatures and many OIDC providers use ECDSA-signed JWTs.
>
> If you have deployed Java 15, 16, 17 or 18 in production, then you should stop what you are doing and immediately update to install the fixes in the April 2022 Critical Patch Update.
>
> Oracle have given this a CVSS score of 7.5, assigning no impact to Confidentiality or Availability. Internally, we at ForgeRock graded this a perfect 10.0 due to the wide range of impacts on different functionality in an access management context. ForgeRock customers can read our advisory about this issue for further guidance.

Closing The Loop

7337 / @7337_Onderzoek

> *sn0863 Use after free.*
> *Why does the deallocated memory not be get zeroed?*
> *Why does malloc not also zero-out the deallocated memory?*
> *Would that not solve the Use after free issue?*

awk / @adrianteri

> *RE: Feedback on #867 Feedback on MS Windows Auto Update Service. My thoughts are on the routers. I've heard Bruce Schneier articulate and give examples on how cheap IOT devices are designed and manufactured in publicly available talks/seminars. A team is assembled and then immediately disassembled after the process and thus there is "NO ONE" left to actively package & push patches to these devices unlike the teams at Apple, Microsoft, Google etc.*

> *Coupled with this, I shudder at the recent incidence and demonstration from #SolarWinds that your supply chain (update servers and processes) can be compromised and commandeered. Schneier comes to the conclusion that market forces **cannot** help here, as consumers want their devices produced **fast** and **cheap** and thus only regulation would be the cure. I wonder what your thoughts are.*

Andy in the UK:

> *@SGgrc on Google auth, email addresses can be checked just by sending an email to it. If it bounces, the account doesn't exist. So it makes very little difference to Google if they reveal the address does exist during the Auth process. However, others using email as ID shouldn't reveal if it's for a valid account.*

Austin / @awestin_

> *Listening to episode 867. During your conversation about authentication processes and email addresses being revealed during the process, Leo asked why an application like Gmail would ask for an email address first before moving on to the next step of the authentication process. This is because many apps use many identity providers and authentication workflows. Your email address will determine which authentication workflow the application will walk down next (i.e. prompt for password, yubikey, redirect you to your custom SAML/identity provider if you have an enterprise account with single-sign on configured, etc.).*

C Me / @CMe06365161

> *Hi, Mr Gibson. I just thought you'd get a kick out of this. On your recommendation I put the McCollum Gibraltar trilogy in my wish list on Amazon. It's only sold as individual books, but wanting to save a few bucks, my son-in-law contacted the seller. That turned out to be the author himself, who agreed to refund the difference in postage. So I got a $7 check signed by McCollum and 3 autographed books for my birthday!*

https://scifi-az.com/ Michael McCollum. The Gibraltar Trilogy / The Antares Trilogy

Sci-Fi

- Finished the Bobiverse... back to Ryk Brown's "Frontiers Saga."
  It NEVER happens recently that I stay up reading after Lorrie falls asleep on my lap... but I could not stop reading this first book of the 3rd 15-book story arc. It's just such amazing story telling.
- Star Trek "Discovery" was too hyperactive, so I have hopes for "Strange New Worlds" next week. It would be great if we could return to Star Trek's original roots, which "The Next Generation" was faithful to, of telling stories rather than only having an excuse for special effects.
- Have not yet looked into the second season of "Picard" after the disappointing 1st season.
- Haven't yet started into the final 6th season of "The Expanse."

SpinRite

The incremental development release of SpinRite which I posted at 1:47 PM last Friday afternoon really surprised me. Tester after tester has been reporting that everything that's been completed so far is finally working perfectly, solidly, and better than ever for them. The reason this surprises me is that the code SpinRite now has is not relying upon any BIOS to interface it to mass storage adapter and drive hardware, yet it is finally working on every piece of hardware that everyone has of any age and vintage.

As we know, operating systems are able to achieve this by bundling a raft of hardware-specific manufacturer supplied drivers which the OS loads on demand based upon the hardware it detects in the system when it's booted. But that's not a practical approach for SpinRite. What I was hoping we would be able to achieve was the creation of universal native drivers, one for IDE parallel adapters with PATA drives, and another for AHCI adapters with SATA drives, where they would simply work everywhere on all hardware from the 1980's through the latest chipsets. I'm not only surprised but I'm also greatly relieved since after many months of work we finally have 100% success and every indication is that this elusive goal has finally been achieved.

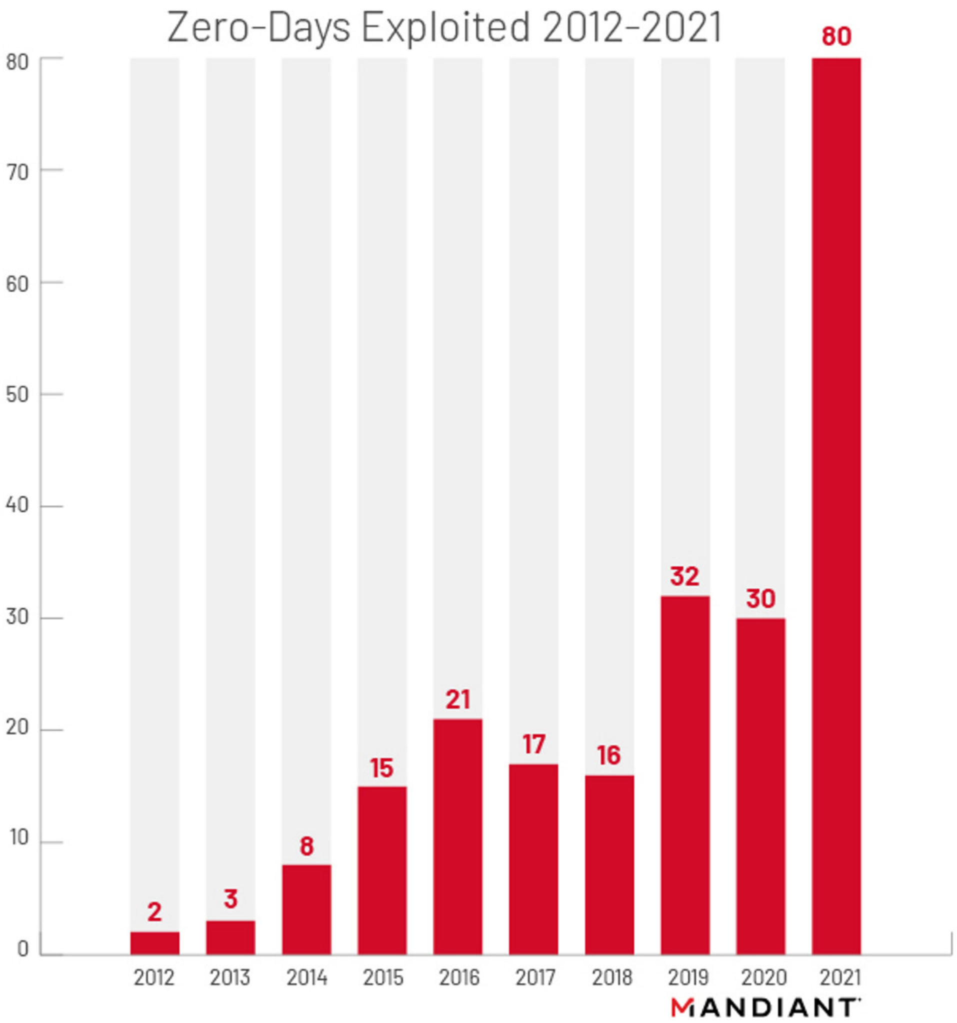
I've been stuck here for a while, perfecting this foundation, because everything that comes next — not just SpinRite v6.1 but v7.0 and all of SpinRite's future — will be built upon it. And it's all based upon the new IO abstraction approach which means that new mass storage technologies – like native support for USB, NVMe and whatever comes next – can be easily added behind the abstraction.

So now, once I catch my breath, I can finish the work on the rest of SpinRite, building upon this new foundation. And then we're going to have v6.1.

The 0-Day Explosion

The most interesting and important class of software and system vulnerabilities are those that are discovered when a security researcher watches something that's not supposed to be possible, happen anyway. Like a specially formed packet hitting a firewall and being admitted through despite the clear firewall rule which blocks its entry. Or the password challenge that is ignored by an attacker who is logged on with administrative privileges anyway. Or the crypto-currency mining malware which suddenly launches, springs to life and begins operating on a system that was just reformatted and reinstalled from scratch. When a researcher watches something that cannot happen, happen anyway, they may have just witnessed and discovered evidence of the exploitation of a previously unknown 0-day vulnerability.

0-day vulnerabilities are a constant topic on this podcast. And just as this podcast appears to be in no danger of running out of security topics to cover, recently posted sobering research from Mandiant and Google's Project Zero teams make it pretty clear that we also won't be running out of 0-days to discuss anytime soon:



Ten years of 0-day tracking. WHAT is going on?

The show notes include Mandiant's sobering 10-year graph showing the number of 0-days discovered each year from 2012 through 2021 last year. The counts for each successive year are: 2, 3, 8, 15, 21, 17, 16, 32, 30, 80.

So, ten years ago, the entire **year** of 2012, we encountered just two 0-days and the next year only three. Then the next three years rose to 8, 15 and 21. Then the next two years dropped back a bit to 17 and then 16 in 2018. 2019 doubled that to 32, 2020 dropped it a bit to 30, but then 2021 exploded from those in 30 in 2020 to 80 0-days just last year. So if you felt as though we had been talking a lot more about 0-day vulnerabilities recently, you would be correct.

Some interesting observations emerged from Mandiant's research, they wrote:

> *In 2021, Mandiant Threat Intelligence identified 80 zero-days exploited in the wild, which is more than double the previous record volume in 2019. State-sponsored groups continue to be the primary actors exploiting zero-day vulnerabilities, led by Chinese groups. The proportion of financially motivated actors—particularly ransomware groups—deploying zero-day exploits also grew significantly, and nearly 1 in 3 identified actors exploiting zero-days in 2021 was financially motivated. Threat actors exploited zero-days in Microsoft, Apple, and Google products most frequently, likely reflecting the popularity of these vendors. The vast increase in zero-day exploitation in 2021, as well as the diversification of actors using them, expands the risk portfolio for organizations in nearly every industry sector and geography, particularly those that rely on these popular systems.*
>
> *Mandiant analyzed more than 200 zero-day vulnerabilities that we identified as exploited in the wild from 2012 to 2021. Mandiant considers a zero-day to be a vulnerability that was exploited in the wild before a patch was made publicly available. We examined zero-day exploitation identified in Mandiant original research, breach investigation findings, and open sources, focusing on zero-days exploited by named groups. While we believe these sources are reliable as used in this analysis, we cannot confirm the findings of some sources. Due to the ongoing discovery of past incidents through digital forensic investigations, we expect that this research will remain dynamic and may be supplemented in the future.*
>
> *We suggest that a number of factors contribute to growth in the quantity of zero-days exploited. For example, the continued move toward cloud hosting, mobile, and Internet-of-Things (IoT) technologies increases the volume and complexity of systems and devices connected to the internet—put simply, more software leads to more software flaws. The expansion of the exploit broker marketplace also likely contributes to this growth, with more resources being shifted toward research and development of zero-days, both by private companies and researchers, as well as threat groups. Finally, enhanced defenses also likely allow defenders to detect more zero-day exploitation now than in previous years, and more organizations have tightened security protocols to reduce compromises through other vectors.*

I thought those points were really interesting. Of course we wonder whether our count of 0-days is recently higher because we're looking harder and more closely for them. This makes sense given that we have established quite clearly that with all software in general, the closer we look, the more problems we'll find. Some of those "more problems" will be exploitable 0-day vulnerabilities.

nd the increasing level of specialization we've chronicled in recent years also leads to higher 0-day counts through the creation of an exploit broker marketplace. Now, those who wish to deploy such exploits don't need to spend their time hunting them down, and those who specialize in hunting for new ways in, can spend all of their time doing nothing else.

And finally, so much of the other lower hanging fruit has been found and pruned that 0-days are becoming the only remaining way to get in. This means that the pressure to discover new 0-days is greater than ever before. And since, as we know, security is inherently porous, the harder you press on it the more results will be obtained.

> *State-sponsored espionage groups continue to be the primary actors exploiting zero-day vulnerabilities, although the proportion of financially motivated actors deploying zero-day exploits is growing. From 2014–2018, we observed only a small proportion of financially motivated actors exploit zero-day vulnerabilities, but by 2021, roughly one third of all identified actors exploiting zero-days were financially motivated. We also noted new threat clusters exploit zero-days, but we do not yet have sufficient information about some of these clusters to assess motivation.*

Just to be clear about that, the primary motivation behind the use of 0-days has historically been state-sponsored espionage. Things like breaking into military contractor’s networks to steal plans for future weapon systems that are still on the drawing boards, and such. But, while such espionage remains dominant by far, last year saw the rise in 0-day enabled so-called “financially motivated” extortion with ransomware and sensitive data exfiltration and threats of exposure.

And Chinese-based cyber espionage groups remains the #1 exploiter of these vulnerabilities:

> *Mandiant identified the highest volume of zero-days exploited by suspected Chinese cyber espionage groups in 2021, and espionage actors from at least Russia and North Korea actively exploited zero-days in 2021. From 2012 to 2021, China exploited more zero-days than any other nation. However, we observed an increase in the number of nations likely exploiting zero-days, particularly over the last several years, and at least 10 separate countries likely exploited zero-days since 2012.*
>
> *From January to March 2021, Mandiant observed multiple Chinese espionage activity clusters exploiting four zero-day Exchange server vulnerabilities collectively known as the ProxyLogon vulnerabilities. Microsoft described activity linked to this campaign as "Hafnium."*

We appear to be focusing upon the right things on this podcast. All of our listeners will recall the attention this podcast gave to the constant stumbling Microsoft was making over their seemingly endless Exchange Server ProxyLogon vulnerabilities. What we didn’t and couldn’t know at the time was that that string of Microsoft missteps was actually translating directly into a string of exploitation with Chinese espionage actors at the other end. They noted that <quote>:

> *While some of the threat clusters involved appeared to carefully select targets, other clusters compromised tens of thousands of servers in virtually every vertical and region.*

This makes sense, right? No one entity owns these vulnerabilities, and we have a heterogeneous environment of uncoordinated groups in China, Russia and North Korea. Some are going to go for high-volume spray attacks whereas others are going to go after specific targets.

And this little tidbit is somewhat worrisome:

> *Chinese cyber espionage operations in 2020 and 2021 suggest that Beijing is no longer deterred by formal government statements and indictments from victimized countries. In addition to the resurgence of previously dormant cyber espionage groups indicted by the U.S. Department of Justice (DOJ), Chinese espionage groups have become increasingly brash.*

The problem is that the world is becoming inured to the whole concept of cyber-espionage, cyber-crime and cyber-attacks. As the years go by and we keep talking about them, it's just human nature that they're going to become less and less frightening and exceptional. They will simply be incorporated into our expectations.

As for Russia, Mandiant noted that:

> *In a sharp departure since 2016 and 2017, we did not identify any zero-days exploited by Russian GRU-sponsored APT28 until they likely exploited a zero-day in Microsoft Excel in late 2021. However, open-source reporting indicated that other Russian state-sponsored actors exploited several zero-days in 2020 and 2021, including possibly targeting critical infrastructure networks with a zero-day in a Sophos firewall product.*

And through the past four years Mandiant said that they had noted a significant increase in the number of zero-days leveraged by groups that are known or suspected to be customers of private companies that supply offensive cyber tools and services.

> *We identified at least six zero-day vulnerabilities actively exploited in 2021, potentially by customers of malware vendors, including one reportedly exploited in tools developed by two separate vendors. In 2021, at least five zero-day vulnerabilities were reportedly exploited by an Israeli commercial vendor.*

Those two separate vendors were the well-known Israeli NSO group and a second smaller and lesser well known vendor of very similar exploit capabilities known as "QuaDream". Like the NSO Group, QuaDream is also Israeli and competes in the same market as the NSO Group, primarily selling to government clients.

Mandiant also noted that, unlike in the past, zero-day exploits were no longer appearing in underground exploit kits. They explained:

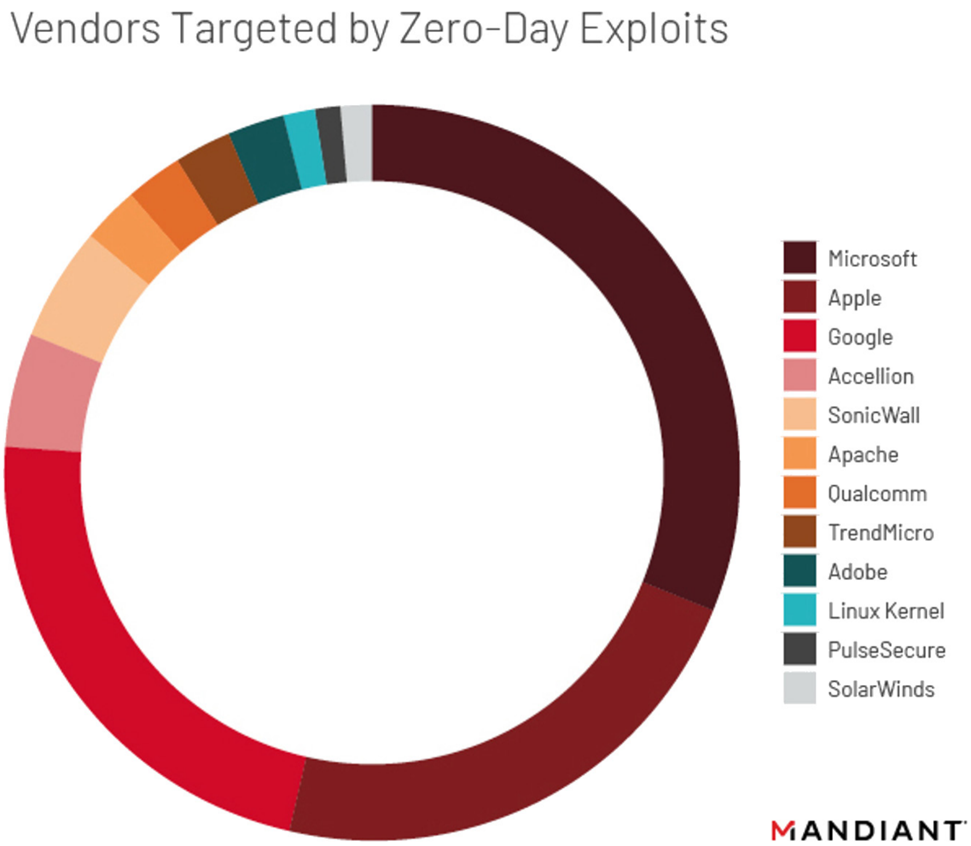
> *Since 2015, we observed a sharp decline in zero-day vulnerabilities included in criminal exploit kits, likely due to several factors including the arrests of prominent exploit developers. However, as the criminal underground coalesced around ransomware operations, we observed an uptick in ransomware infections exploiting zero-day vulnerabilities since 2019. This trend may indicate that these sophisticated ransomware groups are beginning to recruit or purchase*

> *the requisite skills to exploit zero-days that may have been formerly developed for exploit kits.*
>
> *Mandiant has documented significant growth in ransomware in terms of both quantity and impact. Substantial profits as well as the increasingly compartmentalized, outsourced, and professional ecosystem that supports ransomware have provided operators with two viable pathways to zero-day exploit development and/or acquisition: financial resources and actor sophistication.*

In other words, ransomware operations increasingly have the money to purchase high-value, but also high-cost, zero-day exploits from underworld sources. And those underworld sources increasingly have zero-day exploits to offer for sale.

So... where are all of these 0-days being found?

> *We analyzed zero-days from 12 separate vendors in 2021, with vulnerabilities in Microsoft, Apple, and Google products comprising 75% of total zero-day vulnerabilities, likely as a result of the popularity of these products among enterprises and users across the globe.*



So, Microsoft with all of their many products, Apple with their family of iOS devices, and Google with Chrome and the Android platform. Together, those top three account for just a tad more than 75% of those 80 zero-days during 2022. Microsoft has the most, though they also have the most software sprawl. Apple has the next most, with Google the fewest of the three. And given the nature of Chrome and Android, that’s pretty impressive.

There are nine other major and notable sources of zero-days finishing out the Top 12. In order of decreasing 0-day counts, the remaining nine are: Accellion, SonicWall, Apache, Qualcomm, TrendMicro, Adobe, the Linux Kernel, PulseSecure and SolarWinds.

Mandiant noted that:

> *The threat from exploitation of these major providers remains significant, given their prevalence. In addition, we noted a growing variety in vendors being targeted, which can complicate patch prioritization and make it more difficult for organizations who can no longer focus on just one or two vendors as priorities.*
>
> *From 2012 to 2017, Adobe was the second most exploited vendor, with nearly 20% of all zero-days exploiting Adobe Flash alone. We observed a significant drop in Adobe exploitation since then, almost certainly fueled by Flash's end-of-life.*

Yeah, no kidding. How many times did we lament the continued existence of Flash when it was so obviously obsolete while also being such a global menace?

So what is the future outlook for the world of 0-days? Mandiant says:

> *We suggest that significant campaigns based on zero-day exploitation are increasingly accessible to a wider variety of state-sponsored and financially motivated actors, including as a result of the proliferation of vendors selling exploits and sophisticated ransomware operations potentially developing custom exploits.*

In other words, 0-days are big business and that business is currently seeing what can only be described as explosive growth. As for what enterprises can do about this, Mandiant says:

> *The marked increase in exploitation of zero-day vulnerabilities, particularly in 2021, expands the risk portfolio for organizations in nearly every industry sector and geography. While exploitation peaked in 2021, there are indications that the pace of exploitation of new zero-days slowed in the latter half of the year; however, zero-day exploitation is still occurring at an elevated rate compared to all previous years.*
>
> *Many organizations continue to struggle to effectively prioritize patching to minimize exploitation risks.*

Remember that survey we talked about recently where CIOs and IT professionals confessed to just how bad their organizations truly were about applying patches in a timely fashion.

To this, Mandiant added:

> *We believe it is important for organizations to build a defensive strategy that prioritizes the types of threats that are most likely to impact their environment and the threats that could cause the most damage, starting with the relatively fewer number of actively exploited vulnerabilities. When organizations have a clear picture of the spectrum of threat actors,*

> *malware families, campaigns, and tactics that are most relevant to their organization, they can make more nuanced prioritization decisions when those threats are linked to active exploitation of vulnerabilities.*

You know... okay... that just seems so unrealistic. I mean, in a perfect world, sure. But we're talking about an organization dedicating someone to the full time job of essentially continuously surveying the dynamic and constantly changing threat landscape and cross-checking it with all of the organization's potential vulnerabilities. What organization is really going to do that? The truth is that everyone in IT is overworked and there's an awful lot of "hoping for the best" going on.

But there's no argument that, all other things being equal, focus less upon theoretical problems and more on vulnerabilities that are being actively exploited. Mandiant wrote:

> *A lower risk vulnerability that is actively being exploited in the wild against your organization or similar organizations likely has a greater potential impact to you than a vulnerability with a higher rating that is not actively being exploited.* **[Yeah, no kidding.]** *A new CISA directive places a significant focus on those vulnerabilities that are reportedly actively exploited; we believe this will help increase the security posture and strengthen patch management procedures.*
>
> *While zero-day exploitation is expanding, malicious actors also continue to leverage known vulnerabilities, often soon after they have been disclosed. Therefore, security may be improved by continuing to incorporate lessons from past targeting and an understanding of the standard window between disclosure and exploitation.* **[And, of course, we spend a LOT of time here talking about that.]** *Furthermore, even if an organization is unable to apply the mitigations before targeting occurs, it can still provide further insight into the urgency with which these systems need to be patched. Delays in patching only compound the risk that an organization supporting unpatched or unmitigated software will be affected.*

Having read all that and shared all that, and considering the impracticality of expending any great deal of time on prioritization, and also given that low-priority exploits are still exploitable, my own advice to any organization, especially in light of that survey we covered which confessed that patching was clearly not a priority, would be to first and foremost simply fix that. **Period.**

Figure out how to arrange to keep the enterprise's software up to date. Yes, systems need to be taken offline, updated and rebooted. Yes, it's inconvenient. And yes, customers and employees and even upper management in the C-suites will complain. But today's and tomorrow's reality is that last year the number of 0-day vulnerabilities which were being used in the wild exploded from 30 the year before to 80. And those were only the worst of the crop... there were a great many more than those 80 0-days. Microsoft themselves patched 128 vulnerabilities just two weeks ago. It's only going to get worse. Everyone should be as prepared as they can be.