SECURITY NOW!

Transcript of Episode #733

Top 25 Bug Classes

Description: This week we look at the driver behind this summer's comeback in cryptocurrency mining. We also check out a managed security provider's summary of the biggest problems they encounter with their more than 4000 clients. We look at the revised and worrisome update after six years of SOHO router and NAS device security, and we suggest that everyone using Chrome go to Help > About. I found three notes about SpinRite that I'm not sure I ever shared, so I will. Then we conclude with the result of processing the massive CVE vulnerability database which reveals the top 25 most enduring classes of software bug impacting the security of our industry.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-733.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-733-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We recorded this Saturday for a Tuesday airing, but there is so much to talk about, as always: the three most attacked ports, what are they, and what you should do to protect them; why SSH isn't as secure as it ought to be; and why people are still running old versions of Windows. And then the top 25 software vulnerabilities. This is a list every programmer should memorize. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 733, recorded Saturday, September 21st, 2019: The Top 25 Bug Classes.

It's time for Security Now!, the show where we cover the latest security, privacy, and ransomware and breaches with this guy right - right? It's now the Ransomware Breach report with Steve Gibson.

Steve Gibson: It's funny you should say that, Leo, because at the top of the show I talk about how we've pretty much been dominated by ransomware recently. But as a consequence of an interesting driver, there has been a comeback in cryptocurrency mining that we're going to also be talking about.

Leo: Oh, interesting.

Steve: But today is Episode 733 and counting down to 999, which we titled "The Top 25 Bug Classes." The Mitre group that manages the CVE, the Common Vulnerabilities and - it's not exploits. I can't - and something. Anyway, we'll get to it when we get to it. They've processed their massive vulnerability database.

Leo: Exposures.

Steve: Exposures, that's it. I can never remember.

Leo: I couldn't remember it, either. I had to look it up.

Steve: They've processed their massive vulnerability database to - and they do this periodically - to reveal the top 25 most enduring classes of software bugs, which impact the security of our entire industry. And actually, they weren't happy limiting it to 25, so they actually have, like, another 15 afterwards. But we're going to talk about that. Because, I mean, there are, like, lots of problems.

But a managed security provider also that has more than 4,000 small and mid-sized businesses, which they call SMBs for small and mid-sized businesses, has done a summary of all the problems that they continually encounter among their 4,000-plus clients. So that we're going to have fun talking about. We also have a revised and worrisome update after six years. It was in 2013 that a group took a look at 13 consumer routers and NAS devices to see how vulnerable they were. Well, they've updated their report six years later, and the name of their report is just - it's one for the ages. But we'll leave that for when we get to it.

And also we need to have everyone go to Chrome and go Help > About, yeah, Help > About, or no, About > Update. Anyway, whatever it is in Chrome. You need to just go into the About box. I'd been using Chrome all morning and yesterday, and it had not fixed this. There is an emergency five-alarm alert which they're not even talking about because it's so bad. It's a no-click total takeover of the user's system that Chrome affords. And so when you go into Help > About, it'll say, oh, look, I've got an update. And then it'll do it right there. And then restart, and then you're safe.

But until then, and I don't know, like I guess they're just pushing this thing out lazily, but here it is, you know, several - this happened on Wednesday, and so it hadn't happened for me yet. So anyway, lots to talk about. Some fun stuff. I have an interesting graph from this enterprise, this small to mid-sized business company which we will talk about after our first break. And I think we've got a great podcast for our listeners.

Leo: Now, those of you who are eagle-eyed and watching the video might have noticed that the date is Saturday. We're recording this on Saturday because Steve's about to get on a "big ol' jet airliner" and fly to Europe so that he can do some SQRL presentations.

Steve: Yup.

Leo: And then he'll end up in Boston on October 3rd, where we're going to do our special event on the Future of Authentication with Bill Cheswick. Steve and I will be there; Gerry Beuchelt from LogMeIn. It's going to be a great event. I think it's almost sold out. We talked to them, when was that, Steve, on Thursday?

Steve: On Thursday, yeah.

Leo: And I don't know if there's a ballroom big enough at the Intercontinental Hotel to hold all of us.

Steve: Well, yeah, they're running out of room. And in fact this morning I mailed Jennifer 450 SQRL stickers.

Leo: Oh, good.

Steve: So that she could add those to the swag bags that I guess everyone's going to be getting.

Leo: Nice. Yeah, I think we're going to have that many and maybe more. But if you - look. At least, if you want to go, and you're going to be in Boston, 3:45 p.m. October 3rd, that's a Thursday. And we'd love to see you. There'll be an event, about an hour and a half talking event, and then a panel discussion, and then a cocktail party afterwards. You're invited to come and see and speak.

And all you have to do, it's free, is go to twit.to/unlocked, twit.to/unlocked. And that'll be the LastPass page where you can sign up. It's a free event. But here's some cool news. Each of you gets a $100 token to donate to one of three charities. So it's a charity event. So it's kind of a neat way to get 100 bucks to donate. I like that. Twit.to/unlocked.

So we're recording it now. The show will air or go out on the podcast network at its usual time on Tuesday. So if you're listening on Tuesday, it's possible that something has happened, some dramatic event has happened between Saturday and Tuesday. We don't know about it yet. We haven't perfected time travel.

Steve: And that time machine that we were using...

Leo: It's broke.

Steve: It's down for maintenance right now. So we're unable to jump forward in order to provide that in advance in this instance. Well, and you know it was the flux capacitor burned out.

Leo: Oh, no wonder.

Steve: So we're trying to get one from Marty.

Leo: They put a hole in the Stargate, and now we're stuck, yeah.

Steve: Don't you hate when that happens.

Leo: We're stuck in 2019. Who ever thought that would happen?

Steve: It was a problem at the Event Horizon of the Stargate Portal. There was just too much ripple.

Leo: Oh, boy, are we nerds. So, yeah. Anyway, that's why whatever - I know this is confusing, but that's what's going on. And we'll be back on our normal schedule. Oh, I should mention this. Next week is a special kind of evergreen Security Now!. We do a little news, but mostly we're going to talk about Steve's search for the perfect sync solution. So that is going to be really good.

Steve: The Joy of Sync.

Leo: The Joy of Sync. That'll be a week from Tuesday. And then we'll be back on track. And you'll have gone to, where, Dublin, Copenhagen - no, no. Gteborg. Gothenburg.

Steve: I'm speaking to the Dublin OWASP group on Tuesday, and then the Gothenburg, Sweden group on Thursday.

Leo: Wow. Wow. That's great. And so you get a little European jaunt.

Steve: Yeah. And both of those are, like, way oversold, as well, by several hundred. So I think there are going to be some fun presentations. And I added some more schmaltz and polish to the presentation after the Orange County presentation. So I think it's going to be a lot of fun.

Leo: So the schmaltz and polish will not be brought to Boston, however, because we don't have a screen. So it's just you're going to have to talk your way through the schmaltz and polish.

Steve: But I think we still need to do one with a TWiT audience.

Leo: Oh, we're definitely doing that, yeah.

Steve: I would love to do that.

Leo: That's at your leisure, yeah, whenever you want.

Steve: That'll be the schmaltziest that we have because I will be extreme, I will be over-polished.

Leo: Extreme schmaltz.

Steve: I'll have a simonized hard wax shine on it.

Leo: Okay, Steve. On we go.

Steve: So this chart jives with what I expected. And I thought it was very interesting. This was part of the - this is from the report that we'll be getting to a little bit later from this managed security services provider who's got more than 4,000 clients. And the chart is the distribution of Windows operating systems within their client base. And not surprisingly, it shows the number one OS is Win 2008, which of course we know is the server version of Windows 7, at, oh, looks like maybe 33%. About a third of them are that. Windows 7 is in second place at looks like about 30%. Then Windows 2012 at 20%, that's again the server version of, what, I guess that must be Windows 8 server version. And this, again, sort of jives with what I'm expecting.

Windows 10 is down at 5% in the enterprise. And there's even some XP and a little bit of Windows 8. No Windows 2000. That's finally really dead. But interesting also, no Win Server 2019, which of course is the server version of Windows 10. So I think what this is showing us is what we expect, which is that the overall distribution of Windows 7 and Windows 10 is hugely skewed in the end user direction for Windows 10, as a consequence of Microsoft's really strong push in the early days of Windows 10 to end users.

You know, end users are like, oh, new Windows, yeah. And, I mean, as we talked about often, you really had to fight as an end user not to get yourself updated to Windows 10. And of course I famous created Never10 as a little bit of freeware just to help with that process because Microsoft was being so aggressive. So as we would expect, a huge number of end users are on 10.

But also because the enterprise is like, well, you know, we know that when you buy new machines you're just going to get Windows 10. The newest machines it's very difficult to get Windows 7 to even install on them because Windows 7 lacks USB3 drivers. And so you have to - and I do, because I have, because I want Windows 7 on some machines - you've got to jump through some hoops in order to do all kinds of crazy Windows stuff because otherwise the installation just stalls at one point when Windows 7 attempts to run on hardware that has USB3, which all new hardware does. So it's possible to do it.

So I guess my point is that here we're looking at combining Win 2008 and Windows 7. We're looking at more than two thirds of the operating systems in this sample set of 4,000, more than 4,000 small to mid-sized businesses are running some version of Windows 7, which of course is end-of-lifeing at February of 2020. So, and we know that Microsoft has said, well, we're going to make it cost you if you want to stay with Windows 7 by then beginning to charge for extended support past then.

Anyway, it's going to be interesting to see what happens, whether maybe Microsoft capitulates and just says, you know, we really wish people were running 10. But, boy, we acknowledge that nobody wants it, so who knows what's going to happen. Maybe extend free support. Maybe decide they really do need to push people into 10. We'll know here at the beginning of next year. It's going to be interesting to see. But I thought it was interesting to see within that subset of 4,000 clients, two thirds of the operating systems deployed in small and mid-sized businesses were still running Windows 7, again because it works. I mean, it does everything the companies need it to. And so I'm sure they're just deferring it, not that 10 wouldn't. But it's like, let's not mess with it.

So as I mentioned from your perfect segue, Leo, at the beginning of the show, cryptomining has been making a comeback. Obviously we've been focusing a lot on ransomware recently. But it turns out that cryptocurrency mining operations are far from gone. And those who watch cryptocurrency mining have been noticing that it's coming back. The short version of the reason for this is that prices have been going up.

As we know, there was sort of a crash in cryptocurrency pricing back during 2018 from those high-flying days in 2017. In the case of Monero - and I should just note that Monero is the cryptocurrency of choice for those who are stealing cycles from other people's machines. The reason is that Bitcoin's proof of work, and we talked about this back in our Bitcoin podcast before Bitcoin was a thing, Bitcoin's proof of work uses just a simple SHA-256 hash where the challenge is to add something to the blob of data that you're trying to hash to cause some number of least significant bits of the result of the hash to all be zero. That's the challenge is you take the blockchain update, you add something to it, and so you're trying to generate a hash where some number of least significant bits out of the 256 bits the hash produces, some number are all zero.

And what was clever about that was that, over time, by slowly incrementing the number of zero bits that were required, you were able to scale the work required, that is, you kept making it increasingly more difficult to add something to the blockchain chunk that you're signing in order to solve the problem. But the Achilles heel of this approach is that, because the algorithm is just SHA-256, that allows GPUs and then later ASICs to be custom made for ultra high-speed mining of bitcoin.

And of course, as soon as ASICs, you know, Application Specific Integrated Circuits, were available, that upped the ante for bitcoin mining to such a level that commandeered PCs just didn't stand a chance. You just can't compete with a chip which has been custom engineered to just blast SHA-256 in order to solve the bitcoin proof of work. Monero, its proof-of-work algorithm by contrast was deliberately designed to resist acceleration by ASICs, which is why we keep hearing about it being used in hijacked browsers and PCs and servers because it kept the playing field leveled to a much greater degree.

So anyway, over the course of this year, 2019, Monero's price, which had collapsed from its original high of around three to $400, it collapsed down. It lost about 90% of its value during 2018, dropping down to like around 40 to $50. And again, we know that everything for bad guys now is about money. As soon as cryptocurrency became a commodity that you could exchange for currency that you could actually spend, that created a lot of pressure for mining.

Well, what's happened during 2019 is we've seen a gradual recovery, not to the heyday of the flying high days, but to around $115 for a Monero coin. And with that has come increased mining pressure. So anyway, I found an interesting summary of events involving Monero cryptocurrency that I thought I would share, just because it gives you a sense for this.

So in May of 2019 an Intezer, I-N-T-E-Z-E-R, Intezer Labs report described the battle between two cryptomining operations, the Rocke and Pascha groups, who were fighting to infect the same types of Linux-based cloud-based apps. Same month in 2019, like earlier this year, a Guardicore report detailed a Chinese-based cryptomining group that infected over 50,000 Windows MySQL and phpMyAdmin servers in the Nansh0u campaign. And our listeners may remember we talked about that at the time. Also in May, Trend Micro reported that the infamous RIG, R-I-G, exploit kit had started to deploy a Monero miner as its final payload, again because it was beginning to pay again. That cryptominer was aimed at Windows desktop users rather than servers, like most Monero mining operations tend to be, as we know.

The next month, in June of this year, another Trend Micro report detailed a new malware strain called BlackSquid. And we talked about it at the time also. That malware can target both Windows and Linux servers, and also uses additional exploits to move laterally through networks that it's able to get into, to infect as many systems as possible, all to deploy cryptomining payloads. Same month, Trend Micro also reported another malware operation whose final goal was to deploy a Monero cryptominer. Just like BlackSquid, this different malware also relied on EternalBlue to spread laterally through internal networks after compromising an initial point of entry.

Also in the same month, Trend Micro reported details of how the AES DDoS botnet, previously focused on infecting servers to carry out DDoS attacks, had shifted its strategy toward delivering Monero mining because DDoS, maybe if it's DDoS-for-hire it makes money, but not as clearly as being part of a Monero mining pool. So now that's what they're doing. Oh, and they were going after Docker servers as a means of attaining a foothold on a machine on which to run Monero. Same month, June 2019, a security report described another cryptomining malware operation that infected web servers and used a cron job to obtain persistence on infected hosts. And, you know, these are things we have talked about as they have come up in the news over the months.

In June, a Kaspersky report described a new malware strain named Plurox. It targets Windows and comes with several modules for performing cryptocurrency mining in various forms based on the specific nature of the Windows machine that it infects. ESET researchers in June detailed LoudMiner, a malware family that targets both macOS and Windows. And according to ESET, LoudMiner uses virtualization software, QEMU, on both macOS and VirtualBox on Windows, to mine Monero on a Tiny Core Linux VM. Same month, in June, Trend Micro detailed a Monero mining operation during which crooks scan the Internet for Android devices exposing their ADB, which we talked about in the past is the debug port, which then allows them to use that port to plant a cryptominer on unprotected Android machines.

The next month in July, back to Intezer Labs, who detailed the WatchBog cryptocurrency mining botnet which had been operating since late 2018, now had compromised 4,500 Linux machines to run Monero mining. And, finally, last month, in August, a Carbon Black report detailed changes in the activity of Smominru, one of the oldest and largest cryptocurrency mining botnets around. Besides running cryptomining payloads, the botnet also stole credentials from infected hosts, which it later put up on sale online. But cryptocurrency mining was its main focus. So anyway, that gives everyone a sense that, I mean, cryptocurrency mining, because it produces money through exchanges, which are now prevalent, as that price goes up, the pressure to get miners running goes up because it becomes worth the bad guys' time.

So pretty much we're looking now at two major malware campaign systems. We're looking at ransomware, which we've of course been talking about a lot this summer, and cryptocurrency mining, both tied for using cryptocurrency as the common connection. You know, it's no longer the case, as it once was, where viruses and malware just existed because, oh, cool, look, we can propagate across the Internet. No. Now, thanks to cryptocurrency, it's about making money.

BleepingComputer had a headline which caught my eye. The headline was "The Top Three Most Attacked Ports." And I thought, well, that's interesting. I wonder if it's anything we don't know. Well, that story led me to a recently issued 16-page report by a different managed security services firm that we'll be talking about later. In this case, the firm is Alert Logic, that titled their report "Critical Watch Report SMB [Small to Medium-Sized Business] Threatscape 2019."

Their report contains some interesting information and analysis that I thought our listeners would find interesting. They analyzed - these are the guys that have more than 4,000 clients, small to mid-sized business clients, which gives them a really good perspective over what's going on. To produce the report, they analyzed 1.3 petabytes of data, 10.2 trillion log messages, 2.8 billion intrusion detect system events, 8.2 million verified incidents, all across their more than 4,000 customers.

They explained and summarized what they found, as follows. They said: "Small to mid-sized businesses (SMBs)," they said, "are under greater pressure than ever to address cyber threats. Cybercriminals are increasingly targeting smaller businesses in addition to larger enterprises. The principal challenge for SMBs is that they must face these threats with fewer security resources than large enterprises. Limited budgets and staff constraints are causing many organizations to make inadequate cybersecurity investment decisions that continue to put them at risk, but forward-looking SMB leaders are seeking new ways to be 'security smart' as they address cyber risks and respond to attacks."

And of course this is a little self-serving since this group, Alert Logic, they are the people that small organizations outsource their security responsibilities to, to some degree. But it does give them a lot of interesting visibility into a large breadth of organizations and provides some interesting material for us.

They said: "In providing managed security services for over 4,000 organizations, Alert Logic has first-hand insights into how small to mid-sized businesses are being attacked, and the best methods for responding and reducing their attack surface. Since the publication of our last look into the cyber security threatscape in 2018, we've observed a steady increase in attacks and changes in attack methods. Based on an analysis of the" - get this - "5,000 attacks per day we detected across our customer base during the period from November 2018 to April 2019, we identified threat patterns and incorporated those into better defenses for our customers. Additionally, our security researchers actively monitored emerging and evolving vulnerabilities and the attack methods across the threatscape of the open Internet beyond our customer base. This research has uncovered several patterns that specifically affect small to mid-sized businesses, so we've chosen this focus for this latest Critical Watch Report on this SMB Threatscape for 2019."

They said collectively this research is based upon close examination of - and here they talk about their 1.3 petabytes of security data, 2.8 billion IDS [Intrusion Detection System] events, 8.2 million verified incidents and so forth. So they have nine key takeaways. The first, encryption-related misconfigurations are the largest group of SMB security issues. Encryption-related misconfigurations. In small and mid-sized business AWS environments, encryption and S3 bucket configuration remain a challenge. Weak encryption is a top SMB workload configuration concern. They said most unpatched vulnerabilities in the SMB space are more than a year old, which is I think a really interesting and important takeaway.

The three most popular TCP ports account for 65%, so essentially two thirds, of the vulnerabilities in small and mid-sized businesses. So the top three account for the top two thirds. They said unsupported Windows versions are rampant in mid-sized businesses. And I got a kick out of that because of course that takes us back to Windows 7, which will be unsupported in 2020. Outdated Linux kernels, which is something that really hadn't occurred to me before, they say are present in nearly half of all small and mid-sized business systems. Outdated Linux kernels. Active unprotected FTP servers lurk in low-level SMB devices, like IoT things have unprotected FTP servers exposed. And SMB email servers are old and vulnerable.

Let's see if there's anything more. I have some - basically those nine takeaways I also expanded in the show notes. For this encryption-related misconfigurations they said: "Automated patching has made inroads in the fight to eliminate vulnerabilities in the SMB space. Patches are often distributed and can be done automatically across ecosystems. What remains as an issue is misconfigurations which can require remediations ranging from manual reviews to complete architectural redesign." They said: "In our analysis, we determined that 13 encryption-related configuration issues account for 42% of all security issues found."

Talking about AWS, they mentioned that Amazon Web Services - and actually I wasn't aware of this. I was aware that they're a strong player, but they said "with a market share equivalent to that of the next four public cloud providers combined." So, boy, AWS is a heavyweight.

Leo: Well, they're also probably doing more business than all the four combined.

Steve: Right, right. They said: "Our analysis of AWS configuration issues shows that encryption issues affect 33% of the SMB instances we scanned. This indicates encryption is not yet an instinctive behavior," as they put it, "despite being a best practice and a requirement of many regulations including PCI-DSS, HIPAA, HITECH, GBLA, GDPR, NIST," and so forth.

They said that weak encryption remains a continuing problem. And that fourth point, they said most unpatched vulnerabilities - I don't know what's happened to my voice, Leo. It's like yours was at the end of The Tech Guy today. They said: "Most unpatched vulnerabilities in the small and mid-sized business space are more than a year old." They wrote: "Even though automated updates have vastly improved software patching, organizations are still having difficulty keeping pace. When examining the top 20 unpatched vulnerabilities present in the small and mid-sized business space, Alert Logic found that 75% of them are more than a year old.

"The use of open source software, a widespread and established technique for building software projects efficiently, can complicate the patch cycle. This is particularly true when the open source software is embedded. This is a challenge for organizations that leveraged open source resources and libraries. To uncover and reduce the vulnerabilities left by this unpatched code, it is critical for all organizations to invest in third-party validation of the efficacy of the update process in the software development life cycle. Regular vulnerability scanning is a requirement."

So of course we've talked about this often, that the major OSes and now the major browsers have taken this responsibility on. But we're still not seeing this, for example, even in our routers and, to a much lesser degree, in low-end IoT devices. They almost all use Linux micro kernels and various open source resources. And if they've been around for a while and haven't been keeping themselves updated, they can represent a problem.

And on these three most popular TCP ports, they said: "Port scanning is done regularly by both attackers and defenders. Internal security teams, blue teams, can use regular port scanning to help identify weaknesses, firewall misconfiguration issues, and to discover unusual services running on systems. When considering their attack surface, organizations should be aware of which ports have the most vulnerabilities, which is a factor of port popularity more than a statement on the port's relative security."

In examining ports, given that these ports are the ones that are exposed to the Internet, it is no surprise that SSH port 22, that's number one. HTTPS 443, and HTTP 80 made the top three, with 65% of the vulnerabilities. They said: "However, it is interesting to note that the recent MS RDP BlueKeep attack targets the fourth most popular port, RDP 3369." And in fact what was interesting was, in the show notes, under the top missing patches, I pulled a graphic from their report. And it's sort of sobering how many of the patches they show missing are OpenSSH. And that is also the number one attacked port.

So the big takeaway from this is, if your organization is using OpenSSH, it really is crucial to keep it current. Depending upon the device, that has the OpenSSH server running on it, you really want to make sure that it is up to date.

They showed their Table 4 from their document. And I have it in the show notes. OpenSSH Security Bypass is number one. There's an RC4 Plaintext Recovery is number two. Then back to OpenSSH Man-in-the-Middle issue, number three. OpenSSH Arbitrary File Overwrite in fourth place. OpenSSH, a Man-in-the-Middle issue in fifth. OpenSSH User Enumeration issue in sixth place. OpenSSH User Enumeration also in seventh. I mean, it just goes on and on and on. In fact, almost all of the rest are OpenSSH issues. So as we know, OpenSSH is giving very powerful remote access capabilities to enterprises that use it. And it is, when it's secure, it is a very secure protocol. But it's just software, and it's a server, and it is the number one most attacked port on the Internet. So you want to...

Leo: Is an attack as simple as a password attempt, a login attempt?

Steve: It probably qualifies, yes.

Leo: Because I see a million of those on my logs all the time.

Steve: Yes. And so I think you're right. I think it's not necessarily a, you know...

Leo: It's an unsuccessful attack.

Steve: Exactly. So it's an attempt, as opposed to a success at anything.

Leo: Right, yeah. Because I just, by the way, the simplest fix for that is turn off password logins for SSH. Use certificates or use keys.

Steve: Absolutely. Well, I would do that, and I would run on a randomly numbered obscure port. There is no reason to leave SSH sitting on 22. I mean, again, obscurity we know is not security.

Leo: Yeah, I mean, you could scan all the ports pretty quickly, right, if you're looking for - I guess you'd have to - there's 65,000 of them, so it'd take a while. But...

Steve: There are 65,000, and of course there's lots of IPs. So if your IP is not well-known, then that really dramatically increases the scan space. Typically, well, for example, you would not like to be enumerated by Shodan as somebody running an SSH server on port 22.

Leo: Yeah, that's a good point.

Steve: So staying off to the side somewhere probably does make sense. And Leo, let's take our second break, and then I'm going to talk about what they had to say about unsupported Windows versions that was kind of interesting.

Leo: Lord above. Just goes on and on and on.

Steve: Oh, yes. So more than 66% of scanned devices are running Microsoft OS versions that will be out of support by January of 2020. Just, I mean, this is their summary, and this is what I had already noted from the graph that I put up at the top, our Picture of the Week, which I thought was so interesting, you know, Server 2008 and Windows 7. They said: "The current Windows Server release 2019 is almost undetectable"- meaning in terms of its presence among their customer set - "while the majority of devices scanned during the period analyzed are running Windows versions that are more than 10 years old. Additionally, there are still a non-trivial number of Windows XP and even 20-year-old Windows NT devices out there." You know, again, people just - it's running. It's just like, just don't open the closet. Just be careful, you know, just maybe look and see if the lights are still blinking every so often. It's just like they're leaving it alone.

They said: "Even if they are not exposed to the Internet, these targets make lateral movement relatively easy once a host has been compromised. With the discontinuation of security updates and bug fixes for Windows Server 2008 scheduled for 2020, combined with the SMB trend of holding onto old operating systems" - can you say Windows 7, Windows Server 2008 - "this security issue," they say, "is likely going to get much worse next year." I think that's true. I mean, like, bang. This is why I think it's going to be really interesting to see what Microsoft chooses to do about this. I mean, we know it's unreasonable to ask them to continue supporting, except the other thing is you could say, well, if they're going to be supporting anybody for pay, as they are, for what is it, I think three years, three more years where you pay...

Leo: A lot.

Steve: ...additional for each year, yeah. Well, that means that they're doing all the work anyway. So they're saying to other people, well, yeah, we have the updates, but you've got to pay us if you want them. I guess it's reasonable. I don't know. Then they also found, these guys, that outdated Linux kernels were present in nearly half of all systems. They said: "Kernels are the heart of an operating system. They manage everything including hardware, memory, applications, and even user privileges. Kernel vulnerabilities are discovered quite frequently, and fixes are only released for supported versions. In a 2017 article, Computerworld described these outdated Linux kernels as the 'working dead.' Many deployed application systems mask the underlying OS distribution flavor, making it difficult to determine which kernel version is being run." Right? They're just turnkey things, like NAS boxes, for example, that are just like, oh, yeah, don't worry about what's going on here, just we're offering you these services, and they work.

They said: "However, about half the systems we identified are still running a version 2.6 Linux kernel, which has been out of support for more than three years. There are at least 69 known vulnerabilities for this kernel, with many of them relatively easy to exploit, and with 24 of the Common Vulnerabilities and Exposures (CVEs) scoring 7 or above on the severity scale." So that was half the systems still that are out there in their customer base, still running a 2.6 kernel with at least 69 known vulnerabilities, and 24 of those 69 scoring with a CVE score of 7 or higher in vulnerability. So lots of problems that are not being exposed.

And then they talk about FTP being exposed in these environments, and that email servers, of course we've been talking about the Exim problems recently, both that really slow to take over and the new one, the problem in the Server Name Indication (SNI) handshake in TLS, where you just add a backslash null to the end of the server name, and you immediately have a root-level remote code execution compromise. Pow. So, yeah, lots of problems. So anyway, I thought that was an interesting look at, you know, it's not the entire industry, but there's 4,000 customers of a managed security provider which gives us a good snapshot.

Okay. I talked about one of the best named problems I've seen in a long time. There was a Baltimore, Maryland-based security consultancy, Independent Security Evaluators (ISE). This is the group that I mentioned six years ago, back in 2013. They did an analysis of 13 small office/home office, so-called SOHO, small office/home office, SOHO routers and wireless access points, and also some network attached storage devices. At that time they found, that is, six years ago, 57 security bugs, and were able to take over 11 of the 13 devices remotely from outside the local network. Their report was wonderfully titled "SOHOpelessly Broken." So six years ago made a big splash. Lots of popular devices that they tested. We would hope that our beloved industry would have taken this to heart and enhanced its security, fixed all those problems, and - right? No.

ISE, this Independent Security Evaluators group, recently updated its test to what they now call SOHOpelessly Broken 2.0. They've got a whitepaper and a PDF. I've got links to those in the show notes. But I'll just summarize what they found. Once again, they tested 13 devices, some from the same vendors and some new. So they found more than double the number of flaws now as opposed to six years ago. They filed 125 new CVE bugs based on their research. This time around, they were able to obtain remote root access on 12 of the 13 devices. They tested equipment from ASUS, Buffalo, Drobo, Lenovo, Netgear, QNAP, TerraMaster, Seagate, Synology - and by the way, Leo, you'll be glad to know that Synology was perfect, the only one.

Leo: Yay. That's the one I use, yeah.

Steve: Xiaomi, Zyxel, and Zioncom. The attacks included bypassing authentication mechanisms altogether. On one device, the team was able to hijack a cookie authentication system by changing the IP address to 127.0.0.1 and issue unauthorized requests via the authentication API.

Leo: Wow. Wow.

Steve: Basically localhost. And you've got the chart right now up onscreen. The project found that some things had changed since 2013 and others had not. Device vendors had taken steps to protect their software. For example, several used Address Space Layout Randomization, ASLR, to make memory-based attacks like buffer overflows more difficult to successfully exploit. However, they were able to exploit other flaws to bypass the ASLR and still launch the buffer overflow attacks anyway.

So I have in the show notes here a grid. Command injection was the most often vulnerable exploit. The Buffalo TeraStation, the ASUS RT-AC3200, the Netgear Nighthawk R9000, all of these were vulnerable. The TerraMaster F2-420, the Drobo 5N2 was, although I should note that was when the optional web interface was enabled and publicly exposed, which would require some customization. The Zyxel NSA325 v2, Totolink A3002RU, an Asustor AS-602T. The Seagate STCR3000101 was not, the only one along with the Synology DS218 that was not. The QNAP TS-870 was, the MI Router 3 was, and the Lenovo ix4-300d was. That's just command injection.

There were two that suffered from SQL injection. A bunch of them had cross-site script problems. The ASUS RT-AC3200 suffered from a buffer overflow. Five of them suffered from an authentication bypass. Oh, two different authentication bypasses and also the same five in those cases. There was a cross-site request forgery problem. So it looks like a lot of these are web-based problems in web servers that they have exposed and web content that they have exposed. And a bunch of them also had a file upload path traversal problem. And of course, as we know, path traversal is another just ongoing headache for systems. They just are not good about catching those problems.

They said that one device encrypted the PHP files used to process requests through its web interface. But of course, if it's going to encrypt the PHP files, it itself has to be able to decrypt them. So it had to store the decryption key on the device. So they found the decryption key, decrypted the files, and then used those with PHP's system function to obtain shell access.

Paraphrasing from their report, they wrote: "Perhaps more interesting are the number of features that are still missing since SOHOpelessly Broken 1.0. Features such as anti-CSRF tokens and browser security features, which are commonplace now in mainstream web applications, are still rare among the sample devices that they looked at." They wrote: "If vendors had implemented these basic protections, we would not have been able to hack the devices."

ISE tried many different types of attack, as I mentioned in that chart, often chaining them together when necessary to successfully exploit the device. The most successful were cross-site scripting and command injection, which are, as we know, well worn, old, and well-understood vulnerabilities that should also by now be well understood by firmware developers. But in this cross-section of devices, at least, they were not. And they wrote: "Based on their research, Synology comes out on top. Its DS218J," they said, "a device that ISE had also included in the 2013 test, did not show up in any of the broad attack categories, and it had the fewest CVEs at only two: a session fixation bug in its Photo Station application, and the ability to determine metadata of arbitrary files." Which, you know, it's like, okay, who really cares about that?

"Moreover, Synology responded promptly to ISE's bug reports, which isn't something ISE was able to say about all manufacturers." They wrote that some vendors' methods for handling bug reports had improved in the last six years, and others were entirely MIA. Reporting bugs to several of the vendors was not even possible. From some of them, ISE got either no cooperation or no response. Six years ago, back in 2013, none of the manufacturers tested had bug bounty programs in place. Today, Netgear, Synology, Xiaomi, and QNAP all have bug bounty programs in place. So yay for that. That's good news. And I wouldn't be surprised if they're farmed out to HackerOne, which manages these on behalf of companies like this.

So what are our takeaways? ISE's reporting says that, when buying a device, we should look for a history of security vulnerabilities with the vendor, along with how long the vendor takes to fix them. Those can typically be found by searching the CVE repository. We should also avoid using the device with the default configuration. Turn off features that are not needed, especially all remote access features. And please change the username and password from what its default is.

The report also noted that a significant number of devices are deployed, and then never updated afterwards. These devices will be vulnerable, as we know, to any publicly disclosed issues, even if patched firmware is subsequently made available, since they're not going to get the advantage of that. So we need to somehow create some sort of tickler system to remind us to periodically check for updates from that device's vendor and apply them. Some devices, when you log in, they will now at that time, at the time you log in, that kind of wakes them up, and they'll check for an update and, like, give you a warning flag if there is one, or provide some means for manually checking.

But, for example, now would be a good time, like take this opportunity while we're talking about it on this podcast to go check to see if your router has newer firmware because it would be a good thing to update it, if it has. I mean, there's reason they updated it.

The problem is these are by their nature set-and-forget applications. Unfortunately, they are too often set and forgotten. And I'll note that my own advice, and I take my own advice, as I've mentioned previously, is to use a pfSense-based firewall router out on the edge. pfSense firewall routers run FreeBSD. They are turnkey. They have a beautiful web interface, which should not be over anyone's head. I mean, they are very powerful. There's a lot that they can do.

But it is kept current, and it is updated as needed, and it offers every feature you can imagine: OpenVPN. OpenSSH. It has mind-blowing NAT capabilities so you can do things like port shifting in order to get around your ISP that is blocking ports that you'd like to be able to get to remotely. Anything you could ask for. It's 100% open source and free. So you could easily install it in an older machine that's got a couple NICs that you're no longer using.

My favorite drop-in turnkey box is a little Netgate SG-1100 that I've mentioned before, which contains three NICs built in. So you have a WAN and two LANs. That allows you to easily create a truly isolated dual LAN environment with complete network isolation, and then you're able to like, for example, bridge ports only where you want to. For anyone who's interested, I have a link in the show notes. It's not cheap. It's pricey at $180, compared to less expensive hardware that you can set up for yourself if you want to. pfSense.org has a list of all of the hardware that they run on, and they note there that, as pfSense is based on FreeBSD, its hardware compatibility list is the same as FreeBSD's. The pfSense kernel includes all FreeBSD drivers, and the current release of pfSense, which is 2.4, is based on FreeBSD 11.2, which is current.

So anyway, my feeling is - in fact I mentioned, Leo, in our podcast in the future next week, that I have an ASUS router at one location which is doing NAT, and I have it behind a little pfSense firewall because I want all of the fancy features that it offers. And so even if that ASUS router had some security vulnerabilities that existed between the time they were found and published, and I updated its firmware, I've got pfSense as the first device that is on the edge, the public-facing device that deals with the Internet, and I feel very secure with that. So anyway, I just think that's something for our listeners to consider is putting something really strong as the device on the edge. And it doesn't get any better than things running pfSense.

I mentioned the need to go to Chrome About. Chrome got an emergency update on Wednesday. Let's see, that would be Wednesday the 18th of September. They updated to 77.0.3865.90. So you want to make sure you are at least at 77.0.3865.90. Google noted in their Chromium blog, under their release notes, they said: "Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in third-party libraries that other projects similarly depend upon, but haven't yet fixed." I guess that would, for example, be - is it Chredge, the Chromium version of Edge?

Leo: That's what Mary Jo and Paul call it, yeah. The new Edge, yeah.

Steve: That's right, the one we want to be using Edge. So in their blog post they said: "Google has released an urgent software update for its Chrome web browser and is urging Windows, Mac, and Linux users to upgrade the application to the latest available version immediately. It started rolling out to users worldwide Wednesday, containing security patches for one critical and three high-risk security vulnerabilities, the most secure of which could allow remote hackers to take control of an affected system."

Leo: Yikes.

Steve: Yes. But that is with doing nothing. As I noted from their statement above, Google is choosing to withhold details and keep them close to the vest for the time being to prevent hackers from exploiting them and to give most users time to run Chrome and have itself update. And again, I'll note that mine didn't. So I don't know what's going on.

Leo: But you were manually able to do it.

Steve: Yes. All I had to do was go to About > Help, or Help > About. But when I did that, I saw that it said, oh, hold on a second, and then immediately...

Leo: Yeah, you have to restart. Or did it...

Steve: Yes.

Leo: Oh, it hadn't even downloaded it. Oh, that's...

Steve: Yes. It hadn't downloaded it. So it downloaded it, and then it said, okay, we're going to do a restart, and I said yeah. And it did, and then I was up to the latest.

Leo: Do you leave Chrome running in the background? You don't use Chrome, do you?

Steve: No, no, I don't.

Leo: Yeah, so it couldn't update until it ran.

Steve: Well, but I had run it. I had it running, and it had been running all morning, and it had not done that.

Leo: Oh. Oh, it had not.

Steve: Yeah.

Leo: Well, I'm just looking at mine, and it's updating to 76, which is one behind; right?

Steve: Yeah. You want to be at 77.

Leo: Yeah. But I don't keep Chrome running in the background.

Steve: And nor do I.

Leo: So it's going to download, presumably to 76, and then update for 77. I have "automatically update turned on." So I don't know.

Steve: Yeah, they're being a little lazy about it for some reason. I mean, it's the number one browser in the world. And so you can imagine that the update burden is pretty significant.

Leo: Yeah, oh yeah, yeah.

Steve: I mean, it'd be like a DDoS on Google if they tried to have everybody update at the same time.

Leo: Well, it looks like they are because this is unusually slow just downloading 76.

Steve: And you have a good speedy connection.

Leo: I have a gigabit connection.

Steve: That's right.

Leo: It should be all right here. Actually, yeah, gigabit because of the card; 10 gigabits actually to the outside world.

Steve: Yeah. So it's the sending end that is the problem.

Leo: Yeah, must be.

Steve: So there were four problems. They were all Use After Free vulnerabilities in Chrome, meaning that there was some way to use the buffer that had been released from Chrome's memory and could be leveraged to create a remote code execution. Google has paid out a total of $40,000 in rewards to one of the discoverers, Man Yue Mo at Semmle for both of the vulnerabilities, two of the four. And then $20,000 each to discoverers of the other two. So it's nice. They paid out $20,000 per bug in their bounty program.

And they do note that successful exploitation of these vulnerabilities only requires that a specially crafted web page be displayed. They do not require any user interaction. That would mean that a malicious ad that was displayed on an otherwise good website could run code on a user's machine. So we understand why Google is saying, yeah, let's get this pushed out immediately, and why they're not talking about what they've done. They are just making the patch available. And I imagine, given what you and I are experiencing in terms of updating, they're going to be quiet for quite a while, given how bad this thing is. I'm sure the bad guys...

Leo: Yeah. It's finished now. It was on 76. It's now at 77.

Steve: Yay.

Leo: So it took maybe five minutes, yeah. Worth doing.

Steve: And it took you doing it; right?

Leo: Yeah. But I hadn't run Chrome in ages. I'm a Firefox guy, thanks to you.

Steve: Yay, we got you to Firefox.

Leo: You got me. You got me, baby.

Steve: So when I was going through - I have an outline of, like, stuff where I organize and pull all of the show notes together. I use a program I mentioned before that I really love. It's probably now in abandonware. It's called "Thought Manager Desktop." Thought Manager was an app - are you sitting down, Leo? - for my Palm Pilot. And it was a...

Leo: Do you still have a bunch of Palm 7s in the freezer?

Steve: I sure do.

Leo: Oh, my god. Some archeologist is going to find those and really wonder.

Steve: Yeah. You know, once upon a time I thought, well, I love them, so I need to go back to them. But of course that was before the smartphone happened, and that's pretty much...

Leo: Yeah, yeah, everything changed, yeah.

Steve: Yeah. I sure did like the way that handwriting recognition worked. Jeff did just a beautiful job, inventing a slightly modified...

Leo: Once you learned Graffiti, you could do it so fast.

Steve: Yup. Yup.

Leo: It was highly accurate, too.

Steve: Better than keyboards now on our smartphones, I think in many instances. Anyway, so I had this outline. There was a Windows version called Thought Manager Desktop that would synchronize with the Palm. And I just fell in love with it as an outliner. I'm a person who loves outlining. For me, the ability to move things around and organize them, I mean, I was using John Friend's outliner back in the day under DOS. It was PC something. Maybe it was PC Outline. Anyway, the point is that I found, I stumbled upon three SpinRite notes that I don't know I had ever shared because normally I delete them after I put them into a show. And so I don't know if I did or not. So I'll share them now.

One was from Scott in Woodland, California, whose email was alumni.ucdavis.edu. And the subject was "SpinRite Guide for New Hard Drives." He said: "Hey, Steve. I've heard you speak briefly about running SpinRite on brand new hard drives, but what metrics should we be looking at to determine if the drive is healthy or should be returned?" That's an easy one. Run SpinRite, probably on Level 4. At least on Level 2. Level 4 is a little more rigorous. And maybe after an hour or two switch the UI to the SMART page.

SpinRite monitors SMART on the fly. There are bar graphs, horizontal bar graphs which start out at 100% on the various parameters that SpinRite reports. No red should be showing. As SpinRite is running, if the SMART data becomes distressed and depressed, then SpinRite will shorten the green, revealing some red. And certainly a brand new hard drive should not be showing it's relocating sectors. It should not be showing that it's struggling to read. It should not be showing that it's got an excessive number of seek errors.

Those are the three things we generally see in old drives when SpinRite is running on them, when they're beginning to send up smoke signals. Hopefully not actual smoke. We'll get to that in a minute because one guy in Yorba Linda had that happen, apparently.

So anyway, so that's what you want to look for is SMART will show you if the drive is being used. And that's the other thing is SMART data, what we've noticed is, if you look at the SMART data immediately upon starting SpinRite, it'll look all fine. Give it an hour or two, then check it out and see if it still looks okay because it's only when the drive is being asked to do things that it's able to realize it's having trouble. And then SMART says, ooh, ouch, we have a problem. The SMART data tends to be self-healing, that is, it shows recent history, but not forever history. And so, if you aren't running SpinRite, a few months later you'll find the SMART looks good again because you're not asking much from your drive. So it's important to be asking something from your drive.

Speaking of smoke signals, Jeff Gros in Yorba Linda, his subject was "SpinRite is too hot to handle." He said: "Hey, Steve. I'm building a Synology 1815+. I bought six 6TB HGST 7200 RPM drives..."

Leo: Holy cow.

Steve: Yeah, he's building a big monster, "with 128MB cache to run in SHR-2, essentially RAID 6" - which of course is the way I run also - "and a 240GB Kingston HyperX Savage SSD for the Synology to use as a cache. As usual, I run SpinRite Level 4 on all spinning drives and Level 1 on the SSD before putting them to use. My resident SpinRite machine is a 2000s era machine which now runs Linux. However, because this machine is a tower, and the SATA cables are near the top, I have to make my own tower out of books and boxes to get the drive to reach the cables.

"I made the tower out of some boxes with some computer paper toward the top. However, I was worried about the computer paper getting hot and starting a fire. I had some antistatic bubble wrap, so I put this directly under the drive. If it gets too hot, it will just melt the plastic. What could go wrong? Perfectly safe..."

Leo: Whoa.

Steve: He says: "Well, you can probably see where this is going. When I came back to check on the drive a few hours later, SpinRite greeted me with a red screen of death. SpinRite had halted because the drive was too hot. The screen showed 133 degrees Fahrenheit." Then he says: "Not sure if SpinRite dynamically updates this screen" - it does - "or if it was the last reading before halting." He says: "I immediately turned off the machine and detached the drive. Boy. It was hot to the touch. I cooled it as much as possible with some compressed air and let it sit. Now I know why you have heat sinks on your hard drives in that new machine you built. Drives need to breathe.

"The next day I tried again, but gave more thought to my tower, no flammables and no plastic. I also added a desk fan that I use during soldering. Net effect was about 10 Fahrenheit with the fan on full blast, according to SpinRite." Does he mean 10 Fahrenheit? Maybe he means...

Leo: A hundred. Maybe a hundred.

Steve: Or a 10 Fahrenheit gain over...

Leo: Oh, yeah, maybe that's it, yeah.

Steve: Yeah. He says: "The drive seemed to have no memory of going to hell and back." He said: "Steve, thanks for saving my $250 drive from the stupidity of this now humbled SpinRite user. With your help, I now have the Synology up and running." Oh, and he says: "Also, thanks for your recent tip of Afraid.org and the Hover CNAME for DDNS. I'll be trying that out soon enough. Thanks. Jeff." So anyway, Jeff, thank you for the report, in case we didn't talk about it before. And I don't remember ever sharing that story, that harrowing story. So I'm glad I have.

And finally, John Fletcher in Nottingham, England, regarding an SSHD drive. He said: "Having just installed a new Samsung 1TB SSHD with 32GB NAND cache, my thoughts turned to what SpinRite would actually be testing on Level 4, the memory or the platters?" He says: "I've been listening to Security Now! since the first episode. The first hundred" - ooh. Get this, Leo. "The first 149 episodes I listened to in the hospital in one go as I had to remain awake for 24 hours after eye surgery and stay in for a few more days." In other words, he listened to the first 149 episodes back to back in one shot.

Leo: I'm more impressed by the fact that Security Now! keeps him awake because I think it puts a lot of people to sleep. I'm very impressed. It's not the first thing...

Steve: Well, if you're sufficiently geeky...

Leo: Yeah. It's keeping the brain alive. Wow.

Steve: Anyway, John, it's been a couple years, I think, since you wrote this. But for what it's worth, SpinRite is smart about NAND cache. It is able to shut that down on the hard drive in order to bypass it and actually test the platters themselves. So not to worry. SpinRite does what you would want it to do when testing a hard drive that is behind a cache.

Leo: Yeah.

Steve: And on that note, Leo, we are exactly on schedule for our third and final break. And then we're going to talk about what was discovered by processing the massive common vulnerabilities and exposures database to reveal the top 25 most repeated bug classes.

Leo: Cool. Can't wait. All right. Speaking of bugs and flaws and security, the top 25. The big 25.

Steve: So we are all familiar with the CVE, the Common Vulnerabilities and Exposures. We talk about them pretty much every week. It turns out there's also, produced by Mitre.org, is a CWE. It's the Common Weakness Enumeration. Common Weakness Enumeration. They wrote: "The Common Weakness Enumeration Top 25 Most Dangerous Software Errors" - and Leo, there's no surprises here, but it's interesting - "is a demonstrative list of the most widespread and critical weaknesses that can lead to serious vulnerabilities in software. These weaknesses are often easy to find and exploit. They're dangerous because they will frequently allow adversaries to completely take over execution of software, steal data, or prevent the software from working.

"The CWE Top 25 is a community resource that can be used by software developers, software testers, software customers, software project managers, security researchers, and educators to provide insight" - oh, also podcasters - "to provide insight into some of the most prevalent security threats in the software industry. To create the list, the CWE team used a data-driven approach that leverages published common vulnerabilities and exposures, the CVE data, and related CWE mappings found within the National Institute of Standards and Technology, the NIST National Vulnerability Database, the NVD, as well as the Common Vulnerability Scoring System" - boy, we're in acronym land here - "the CVSS scores associated with each of the CVEs. A scoring formula was then applied to determine the level of prevalence and danger each weakness presents. This data-driven approach can be used as a repeatable scripted process to generate a CWE Top 25 list on a regular basis" - what fun - "with minimal effort."

So the number one ranked most, what is it, Common Weakness Enumeration most widespread worrisome class of bugs, to no one's surprise they name Improper Restriction of Operations Within the Bounds of a Memory Buffer. In other words, the memory overrun, buffer overflow, whatever you want to call it. It was to give some people a sense, because this falls off relatively quickly, this was at 75.56 was the score. So a 75.56 score for basically buffer overrun.

Number two, at 45.69, so look at the size of that drop, even in second place, meaning that buffer overruns, they still rule the roost. Number two, Improper Neutralization of Input During Web Page Generation, in other words, cross-site scripting. So Improper Neutralization of Input During Web Page Generation, so that is taking user input and, as they call it, "neutralizing it," meaning sanitizing it, purifying it, maybe just rejecting it, depending upon what it's got. But that's still, that's number two on the hit list of problems.

More generically, and therefore at about the same level, 43.61, they call Improper Input Validation. So again, it's a problem that developers continue to have, which is assuming that users are going to behave themselves, assuming that everybody using a web form wants it to work the way it was designed to. Well, by definition, bad guys want to break that assumption. So it is really important that input gets validated. Fourth is Information Exposure, sort of generically. And that's got a score of 32. Out-of-bounds Read comes in at about 27.

Number six, Improper Neutralization of Special Elements used in an SQL Command, in other words, SQL Injection. So that's the formal term for SQL Injection, Improper Neutralization - there again, they like the term "neutralization" - of Special Elements used in an SQL Command. So of course we know what that means. It's too often the case that the system will interpret as an SQL query something - or an SQL command, something that the user inputs. So that's sort of a subset of not doing proper input validation.

And then coming in seventh is those three problems that we just had with Chrome, Use After Free problems, where you still have a means of having a handle to a buffer which has been released to the operating system, and you're able to get up to mischief with that. Then number eight - that was at a rank of 18, or seventh in ranking, but a score of 18. Then we have the Integer Overflow or Wraparound, which is now - so that's less, this is less of a public exposure problem and more of an internal bug problem, where as a consequence of an integer wrapping around from its maximum value to its minimum value, which happens where you're forced to store integers in a fixed-sized field, that's a bug, and it can be leveraged to create problems.

Number nine, Cross-Site Request Forgery, CSRF, which is another way of abusing web pages on web-facing applications. Number 10, Improper Limitation of a Path Name to a Restricted Directory. And then that's sort of their formal term for path traversal. And we've often talked about the problem that that creates for people.

Number 11, and here again we have neutralization: Improper Neutralization of Special Elements Used in an OS Command. So that's OS Command Injection. Once again, doing something where, for example, that was the bug in the Exim server that took a week to exploit was it was possible to use an exec function to get the Exim server to pipe OS commands where the field itself was like a reply-to field in email that should never have been executable. So that would be improper neutralization of special elements used in an OS command.

Twelfth position, Out-of-bounds Write, where you are writing where you're not supposed to, one way or another. Thirteen, kind of generic, Improper Authentication. That would be the problem that we've been having with the RDP servers recently because they were saying, hey, we're here, waiting for someone to connect on port 3369, you know, or was it 3389? Anyway, I think it was 89. And so unfortunately there was a bug in authentication that allowed anybody to do that.

Null Pointer Dereferencing, that typically causes a crash. The idea is that a pointer is something which points to something else. And so dereferencing a pointer is the act of accessing the thing that the pointer points to. It's often the case that pointers have a null value in them, unfortunately. And so when you try to access the thing the pointer points to, you're accessing address zero, which is almost always a bad idea and causes bad stuff to happen, crashing and so forth. So that's ranked 14. Although now we're down to a score of 9, where we started at 75. So, or actually 9.74, so about 10.

Fifteenth rank, Incorrect Permission Assignment for a Critical Resource. Yeah. So you basically haven't properly restricted access. Sixteen, Unrestricted Upload of a File with a Dangerous Type. Yeah, don't do that. That's not good. Number 17, Improper Restriction of XML External Entity Reference. Again, XML gives people a lot of power. You want to make sure that you've got it under control. Eighteenth in rank, Improper Control of Generation of Code, in other words, Code Injection. Use of Hard-coded Credentials. Whoops. That's 19th. Uncontrolled Resource Consumption. Yeah, that's not good. That's 20th.

Missing Release of Resource after Effective Lifetime. So that's probably a problem with garbage collection, where you have not released something after you no longer need it, typically considered garbage collection in automatic languages. And that can be a problem. Untrusted Search Path is in 22nd position. Deserialization of Untrusted Data. Whoops. It's interesting that that's as low as it is. We've talked about this. Deserialization is the process of basically interpreting the serialization of a complex object like a JSON blob. JSON can be serialized. And when it's deserialized back into the data structure, normally the interpreter trusts the serialization. But that trust can be abused. And deserializing untrusted data has been a source, as we've talked about before, of all kinds of exploits. Twenty-fourth position is Improper Privilege Management. That's never good. And then 25th, Improper Certificate Validation. You want to validate your certs, otherwise what's the point of having them?

And then they weren't happy, as I mentioned at the top of the show, stopping at just 25. So they had a section: Weaknesses on the Cusp. And they said: "As in years past, the CWE team feels it's important to share additional CWEs that scored just outside of the top 25. Per the 2019 scoring formula against the NVD dataset, these weaknesses were potentially not severe enough, or not prevalent enough, to be included in the 2019 list. Developers that complete mitigation and risk decision-making on the 2019 CWE Top 25 may want to look for these other weaknesses potentially present in their software. For these reasons, users of the 2019 CWE Top 25 should seriously consider including these additional weaknesses in their analysis." And I'm going to run through these just very quickly. Here we have the Loop with Unreachable Exit Condition. And Leo, of course that's more colloquially known as the "endless loop."

Leo: Endless loop.

Steve: The infinite loop. Yes. And Windows users are used to seeing the "application not responding" message when that happens. We have Insufficiently Protected Credentials. Incorrect Type Conversion or Cast, yeah, that's actually - we've had a couple of security problems over the years where that was the underlying vulnerability.

Leo: I'm surprised that's not higher because casting...

Steve: I am, too, yeah.

Leo: Casting is always a problem, yeah.

Steve: Yeah. I am, too. Concurrent Execution using Shared Resource with Improper Synchronization. That's the formal term of the Race Condition, which is also something that happens all the time. Server-Side Request Forgery, as opposed to client-side. But that's there. Double Free. You only want to free your memory once. You don't want to free it twice because after the first free it's free.

Leo: There's nothing there, baby.

Steve: When you try to free it again, all kinds of bad things are going to happen. Then we've got the URL Redirection to an Untrusted Site, known as Open Redirect. That's not good. Incorrect Authorization. Missing Authorization. That's worse than incorrect. Inclusion of Sensitive Information in Log Files.

Leo: Oops.

Steve: Yeah, that's bad. Missing Authentication for Critical Function. Session Fixation. Inadequate Encryption Strength. Allocation of Resources Without Limits or Throttling. Ooh. And a Reachable Assertion. And I have no idea what that is, a reachable assertion.

Leo: That's from unit testing, maybe, where you assert, try - you try code, and then if it's...

Steve: Yeah.

Leo: Yeah, I don't know. That wouldn't be necessarily a bug. It would just be a test that...

Steve: Yeah, I know. May all your assertions be reachable.

Leo: Yes. Huh. They have a much larger description of each of these. It says this product contains an assert or similar statement that can be triggered by an attacker, which leads to allocation exit.

Steve: Ah. Oh, a reachable.

Leo: Reachable.

Steve: Ah, a reachable...

Leo: So in other words, your tests should not be reachable.

Steve: Yes. Turn those off when you ship the code.

Leo: Yes. While assertion is good for catching logic errors and reducing the chances of reaching more serious vulnerability conditions, it can still lead to a denial of service if you can get to it. Wow. That's good. Every programmer should read this and memorize it and put it on the wall.

Steve: Yeah.

Leo: Because these are just easy mistakes to make, and it's good to be thinking about these.

Steve: Well, and it's so nice to, I mean, like when you write some code, then just sort of go over the list and check it in your head. You know, like just sort of cross-check what you write with this list. And if you do, the problems you have will not be on the list.

Leo: Yeah. Well, also it's a great way to debug, like did I do this? Did I do this? Did I do this? That's great.

Steve: Yeah. It's a checklist.

Leo: Should be in your head. I love it. That is really actually a great list. I'm going to print that out, put that on the wall. Very useful stuff.

Steve: Well, and what's interesting, too, is it wasn't some guys that sat around and made up the list. It wasn't a committee.

Leo: This is real.

Steve: It was 100% data driven. They analyzed, I mean, they used a scripting process to analyze the preponderance of CVEs. And what came out of it has no surprises. It's exactly what we would expect.

Leo: Yeah. Yeah. Not a surprise at all. Which even underscores further the truth that every programmer should be well aware of these because these are mistakes that keep getting made. Wow. Great. Good stuff. So somebody in the chatroom was saying does Mitre still maintain the CVE?

Steve: It's still there. Anybody is able to apply for one in order to actually get a designation. But it's at Mitre.org is where that lives.

Leo: Okay. He thought they'd moved on. Somebody else had acquired it. But maybe not. Steve, what a good one. A great - this is an evergreen, too. Those are bugs everyone should know and memorize. And we all do.

Steve: Unfortunately, they will still be relevant five years from now.

Leo: Yeah, yeah, it's evergreen.

Steve: Because they show no sign of actually going away anytime soon.

Leo: No, no, no. We do this show normally - we're doing it Saturday because Steve's about to head out to Dublin and Gteborg, Sweden. He's going to have a great time showing off SQRL. By the way, I haven't mentioned this, but anybody who downloads the show notes already knows it, but the new end of show notes logo is a little SQRL. It used to be a 30. But now it's a little SQRL. I like that. Do you put that everywhere?

Steve: Yeah.

Leo: It's on your mic flag. It's everywhere.

Steve: I don't yet have a tattoo on my butt the way you do, Leo.

Leo: That's coming soon. There's some good people there in Gteborg who could do it. If you get a chance to see Steve there, or in our upcoming event in Boston, you've got to. We'll be talking about SQRL in Boston, and I know he's going to be - that's the whole presentation in Sweden and in Ireland. But of course you can come back here every Tuesday once Steve gets back. So you're going to be gone - this coming Tuesday you're going to be gone. The following Tuesday you're going to be gone. We'll be back doing the show October 8th. But we will not miss an episode because that's Steve's...

Steve: Well, for what it's worth, I just should mention that the show notes will be delayed because I'm not making any attempt to do them long distance. I'm shutting everything down. I'm locking down my security. I'm just not going to attempt to do that remotely. So I will catch up with all the show notes. I'm sure that - I talked with Elaine about this. She's going to grab the podcast and transcribe it. So everything will be waiting for me when I return, and I will then catch up.

Leo: Nice. You go to GRC.com to find all that, the Gibson Research Corporation. That's where Steve hangs his hat on the Internet. It's also where SpinRite lives, Steve's bread and butter, the world's best hard drive maintenance and recovery utility. Get it at GRC.com. Everything else is free, including ShieldsUP!, including SQRL, including everything. And it's a lot of fun. It's one of those sites you go to and you end up five hours later going, where did the time go? There's so much stuff there.

GRC.com. Look for 16Kb or 64Kb audio versions there, the transcriptions. We have audio and video at our site, TWiT.tv/sn. And of course you can always subscribe on your favorite podcaster. That way you get it automatically. Just subscribe to Security Now!, and you don't have to worry about when we're recording it. You will get it at an appropriate moment in the space-time continuum. Your fabric will not be rent.

Steve: Yes. We will straighten all that out for you.

Leo: Yes. All the wrinkles in time, flattened. Steve, have a great trip. Man, I'm not going to see you till the other side. I'm sure you're going to have a blast in Ireland. I'm sure there'll be a few people trying to buy you a drink at the pub there.

Steve: I've heard that happens.

Leo: I think it might. Think it might. But we'll see you next time on Security Now!.

Steve: Okay, my friend. And I'll see you in Boston.

Leo: Bon voyage. Yeah, I'll see you in Boston.

Steve: Bye.