Minimum Viable Secure Product

Description: When is it far better for a security researcher to just keep their mouth shut? Are all Internet-based secure note exchanging sites created equal? What's been happening in the lucrative and slimy world of zero-days for pay? And what has NASA just learned about the state of Voyager 1? Something momentous has happened with SpinRite, and we're going to take a deep dive into an important industry initiative that just acquired an important new contributor.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-969.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-969-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got a great show planned for you. Coming up we're going to say hello to V'Ger, Voyager 1. There's some really interesting news in attempting to reestablish communications. Why is it better sometimes for a security researcher to keep their mouth shut? Steve has some strong words. And we'll talk about the slimy world of zero-days for pay. Turns out you can make a lot of money if your ethics are a little, you know, fuzzy. All that and more, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 969, recorded Tuesday, April 9th, 2024: Minimum Viable Secure Product.

It's time for Security Now!, oh, yes, once a week, a must-listen, Steve Gibson and the latest security news. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again for another, well, we've got a bunch of really cool stuff to talk about.

Leo: Oh, good.

Steve: The podcast does not have a particularly catchy title. I guess I would argue that the podcast that we titled "1" was probably...

Leo: Was catchy, yeah.

Steve: That was catchy and pithy and all that.

Leo: Pithy.

Steve: This one is Minimum Viable Secure Product. Which I had fun with it, mostly because it is so dry. But it's also very important. So that's how we're going to - we're going to do a close look at an important industry initiative, which is what this is, titled unfortunately - you can just say MVSP. That's, like, better, right, MVSP. But it's Minimum Viable Secure Product, which just acquired an important new contributor.

But first we're going to look at when is it far better for a security researcher to just keep their mouth shut, and what happened in this case when he didn't? Are all Internet-based secure note exchanging sites created equal? What's been happening in the lucrative, if slimy, world of zero-days for pay? And what has NASA just learned about the state of Voyager 1? That little intrepid puppy just - oh, and by the way, Leo, I watched the documentary you referred to, "It's Quieter in the Twilight"?

Leo: Oh, isn't that a great - all about V'Ger? I loved that.

Steve: It was really, really fun, yes. Also I will tell everybody that something momentous has happened with SpinRite, and then we'll take our deep dive into the fascinating - actually it is because it's going to make a lot of our listeners smile, Minimum Viable Secure Product, because it encapsulates the history of this podcast in many senses. So I think - and of course a great Picture of the Week. So for Podcast 969 for April 9th, here we go.

Leo: Love it. Here we go. 969 is on the air.

Steve: We have a picture. So I've had this one for a while. I've mentioned having a bunch that were in the queue. I got a kick out of this one because - so we have a one-story building, sort of it's like the side of maybe it would be a strip mall or something, where you have, you know, the need for roof access, but it's in sort of a public environment, so you just don't want random people climbing up on the roof; right?

Leo: Climbing up, yeah.

Steve: Yet you still need, like when one of the AC units dies or something happens, you need service people to get up. So you've got a ladder running up the side. Now, okay. How do you lock the ladder?

Leo: [Laughing]

Steve: And you've just seen it.

Leo: Yes. Go ahead. So this is clever, in a way.

Steve: Uh-huh. So how do you lock the ladder? You know? Ladder's got rungs. I know. It's great. So some clever individual figured, okay, we're going to put a hinged sheet

metal panel, like across the front of this ladder. And we're going to lock it with a padlock, and only the manager of the complex has the key for this padlock.

Leo: Brilliant. Brilliant.

Steve: And, yeah. So, like, so basically it's completely closing off the rungs of the ladder.

Leo: Yeah.

Steve: So, you know, you can't climb up a smooth surface.

Leo: No.

Steve: It just, okay.

Leo: Can you? No.

Steve: Now, unfortunately, the ladder is mounted to the side of the building with a set of rungs.

Leo: That are exactly ladder-like.

Steve: So, gee. If I wanted to climb up on the roof, how would I do it? Well, I'd just go up the side of the ladder where the rungs are mounted to the side of the building, until I get above the sheet metal sheet. Then I can switch over and climb up the rest of the ladder. So...

Leo: Wow. Well, they handily give you a ladder on the ladder, which is great.

Steve: Yeah. We have here a failure of imagination.

Leo: Or malicious compliance; you know? The guy who installed it said, I know this is useless, but that's what they're paying me to put in, so...

Steve: This will serve you right, folks.

Leo: Serves him right.

Steve: Yes. Thank god it's got a padlock, that's all I can say.

Leo: That's hysterical. Oh my god.

Steve: Wow. Okay. So what year is it, Leo? It's 2024; right?

Leo: 2024, I believe, yes, sir, yes.

Steve: Okay. It's still the case that publicly accessible, Internet-connected, high-profile devices are being found to have manufacturer hardcoded remote access credentials. Now, I suppose that would be more difficult for us to believe if it wasn't so long ago, like it wasn't that long ago is what I meant, that Cisco's own internal audit of their devices, which was spurred - and we covered this on the podcast - by repeated discovery of their own hard-coded credentials in their equipment, kept turning up instance after instance of the same thing. So like it's not shocking because, you know, here even the market leader at the high-end enterprise-grade networking gear was suffering from the inertia of the way they'd always done things before. Even a decade after, it had really no longer been safe to do that.

Okay. So it's taken some time. But as we've been discussing more recently, we finally have - we're kind of coming to an agreement, and we're going to see more of this agreement here at the end of the podcast, on a set of straightforward and widely recognized best practices. So I've just put out five of them here.

No default manufacturer-set credentials of any sort, anywhere. Right? The first time a device's firmware boots, it should see that its credentials are blank. After it's had the chance to generate sufficient internal entropy for random number generation, it should create a very strong new default credential from scratch and present it to its user. Now, the user will be free to then change it to whatever they want. But by default, any newly installed device will give its own security a head start by defaulting to something strong. And nothing from the manufacturer.

You know, yes, if it makes technical support more difficult, that's just tough. Once upon a time it might have been fine to default the username and password to "admin" and "admin," and to print that on the quick start guide. But those days are long past. All of our listeners have seen those days. Security is inherently porous, and the pressure against our porous security is now steadily on the rise. That's the other thing that it is like so clear now. Okay. So first, nothing by default.

Second, and this was one of the new things that has surfaced recently, proposed by CISA, that I really think is right, and that is physical access required. Changing the security of anything in any dangerous direction, like turning on UPnP, which should certainly not be on by default, including that administrative username and password we were just talking about, must be accompanied by a physical button press, either beforehand to enable admin access for some period of time, or afterward to confirm the application of the new settings into the current setup.

And yes, again, that will make fully remote admin impossible. But it will also make fully remote admin by bad guys impossible. Right? And we've seen over and over again that it's not actually possible to have one without the other. So yes, it'll be inconvenient. Yes, you may have to call down to the basement and tell Morris to press the button now because you want to apply these settings from the comfort of your office on the 20th floor. Fine. Do what has to be done. But it will keep the bad guys out.

Number three, only absolute minimum functions publicly exposed by default. A router, for example, must have its WAN interface publicly exposed to be at all useful as a router.

But it does not need a single additional service beyond that. So therefore not one additional service should be bound to that interface, the WAN interface, without explicit manual enabling accompanied by clear caution notices and, yes, pressing a physical button on the router. And needless to say, when that service is instantiated, it will start off with a fully random and strong username and password. You know, we know how to do this now.

Four, autonomous firmware updates received and applied by default. All connected devices must ship from the factory with their auto-firmware-update system enabled by default. Again, it could be turned off. But it should, everything should default to most secure. These devices should then periodically ping a factory server to check for the availability of new firmware. The router's ping should include its current firmware version, and the cryptographically signed ping reply should provide the router with the current version and the urgency associated with moving from where it said it is to the latest today. The device's admin can decide which levels of urgency are permitted to auto-update. Maybe you want to back off and only allow the ultra super emergent updates to happen immediately. Otherwise they might need management oversight, or they might be deferred until 2:00 a.m., whatever. So there could be administration over that, but the system ought to take care of itself.

And, last, firmware for devices must be maintained as long as they're in service. And this is probably the trickiest of the five because you have to then solve the problem of how long should a supplier be reasonably responsible for making the firmware of an end-of-life and out-of-service device current? You know, this is a decision that is reasonably up to each supplier. But because it affects the buying decisions, whatever the supplier's decision may be, it must be clearly and publicly stated so that prospective purchasers can plan accordingly.

Okay. So we have five fundamental principles: No default credentials, physical access required to apply any configuration changes that might be dangerous, no unnecessary services enabled, auto-firmware updates enabled by default, and a clearly stated end-of-life ongoing support commitment. So if you think back through all of the individual events, many of them having significantly catastrophic consequences that we've covered through this podcast's 19 years, almost every one of them could have been prevented if these five fundamental principles of safe Internet connectivity had been in place.

So today, as luck would have it, we have news of yet another: A researcher discovered that D-Link Network Attached Storage, NAS models DNS-320L, 325, 327L, and 340L all share a distinguishing and, I don't know if it's horrifying or it's disappointing. Anyway, he found that the firmware they share had been hard coded by D-Link with the same, fixed, remote access backdoor username and password. And even the term "password" is being generous here because it's left blank. His GitLab handle is NetSecFish, obviously as in "Network Security Fish," and his page on GitHub says: "The described vulnerability affects multiple D-Link NAS devices," and then he enumerates those four.

"The vulnerability lies within the nas_sharing.cgi URL, which is vulnerable due to two main issues: a backdoor facilitated by hard-coded credentials, and a command injection vulnerability via the 'system' parameter. This exploitation could lead to arbitrary command execution on the affected D-Link NAS devices, granting attackers potential access to sensitive information, system configuration alteration, or denial of service by specifying a command."

Okay. This guy's subsequent Internet search revealed 92,589 of these devices currently exposed on the public Internet.

Leo: Oh, man.

Steve: Yes, 92,589, right now on the Internet, with the highest concentration appearing at IP addresses in the U.S. The big problem is that this family of D-Link NAS devices were first phased out of D-Link's product line on October 29th, 2017. And two years later, on the same day in October 2019, all support for them was terminated by D-Link. Even so, it appears that our Network Security Fish gave D-Link no prior notice of his discovery before he published it, no opportunity to decide whether they wished to alter their standing policy for out-of-service-life products. Two weeks ago, on March 26th, a date which I'm fond of because it's the day I was born...

Leo: And Leonard Nimoy.

Steve: Leonard Nimoy and I. He simply published everything he knew about this. Today, his public GitHub page thoroughly documents - today and for the last week, or the last two weeks, publically documents how a simple HTTP GET query, containing the username "messagebus" and a blank password, can be used to cause any of these 92,589 currently online network attached storage devices to execute arbitrary commands. We see from the packet capture he enclosed that he issued - I have it in the show notes. He issued a command to a machine at a redacted address - and even doing that would be controversial unless the machine was one he owned, which seems unlikely - a command to which the machine replied with a valid HTTP 200 OK response. It included other reply headers and XML in the body of its reply. So from that we learn that the D-Link NAS devices are running the "lighttpd" web server v1.4.28.

Now, you know, I'm unhappy with what D-Link has been found to have done. We'll get to them in a minute. But it's not clear to me that anyone is helped by Fish's public disclosure, and that 92,589 people and their networks and their data stand to be attacked and compromised as a result. You know, before he made this needless public disclosure, it might have been that someone might have eventually stumbled onto this, too. And it's true that we don't know that it hasn't happened already. But we do know that it was never public before, and that now it is.

And another lesson we've learned through the years of shared experience on this podcast is that the height from which the fruit is hanging matters a lot. Far more low-hanging fruit is plucked than high-hanging fruit. There is an endless supply of script kiddies able to issue a series of simple HTTP GET queries. And how are any of them going to be able to resist any target as tempting as this? This is the definition of a script kiddie-class vulnerability. So this person's disclosure two weeks ago has, without any question, compromised the security of 92,589 still online, still functioning, still in use, and still sitting on other people's networks, D-Link network attached storage devices.

At the same time, a fair question is "What would Google do?" And by "Google," I'm referring to Google's Project Zero with their famous 90-day disclosure policy. Now, in fairness, he didn't give D-Link even a one-day. This was a zero-day disclosure. But in the case of Google with Project Zero, you know, they give manufacturers 90 days to produce a patch for a discovered flaw before Google will release the flaw's technical details. And as we know, many companies have felt significant pressure to fix a vulnerability that Project Zero had found in their products.

But in this instance, the answer to the question "what Google would do" is that I'm not 100% sure. For one thing, these are long-out-of-service devices that are no longer being maintained by their supplier. And we know that even Google has, like, known vulnerabilities in Android that is no longer being maintained on Android smartphones. And it's like, well, sorry, you know, we're not patching that version of Android any longer. So that's a policy. But the other thing is this is not actually a bug. This is an

undocumented deliberate backdoor that D-Link embedded into their devices, their consumer network NAS devices, for some unclear purpose. So because it's old, out of service, and not a bug, I'm pretty certain that Google's Project Zero would never even consider taking this up in the first place. And if they knew about it, they would just be mum.

So this brings us back to D-Link, a well-known, reputable, popular, Taiwanese network hardware manufacturer that's been around since 1986. I've purchased a bunch of D-Link equipment, mostly network hubs, I think, through the years. So I suppose that at least among those of us here and others who learn of this past behavior, this at least tarnishes their brand in our mind. Right? I'm like, I'm not buying a D-Link NAS now when I know that for some reason they've got this undocumented backdoor. They did. We don't know about them today.

So with devices that were discontinued six and a half years ago, that went out of support four and a half years ago, we would have to assume that they have, even if they wanted to, no way of remotely updating their firmware. You know, that's like a thing that we're beginning to see now on new devices because it's been determined that that's the best practice. If the owners of those devices had registered them, well, then, D-Link could conceivably today reach out to them. But what would they say? "Uh, you know that NAS that you purchased from us 10 years ago? Well, uh, we had planted a secret remote access backdoor into it, and it was recently discovered and has now been made public. So, you know, we're really sorry about that, and you should probably definitely immediately disconnect your D-Link device from the Internet. Have a nice day. And would you like to consider buying a newer NAS from us? We have a bunch of nice shiny ones for you to consider." Well, right. So they're not going to say anything.

And I don't know that they're even legally vulnerable here. So it's not clear that anything could be done on the legal front. This was a design decision made by D-Link. Bad, in retrospect, but there it is. And they may have some justifiable rationale for it. The username "messagebus" kind of suggests that perhaps these devices could be set up in a cluster, where this "messagebus" allowed them to interoperate in some fashion. So, you know, should have never been put on the public Internet. Maybe that was a mistake. Maybe it's only meant to be on the LAN, and somehow that wasn't done right. In other words, bad design not to control who could use it on which interface, but it wasn't an obvious crime on D-Link's part.

And speaking of crime, here it comes. Yesterday, entirely predictably, Ars Technica's headline read "Critical takeover vulnerabilities in 92,000 D-Link devices under active exploitation." Now, that didn't take long. Ars wrote: "On Monday," meaning yesterday, "researchers said their sensors began detecting active attempts to exploit the vulnerabilities starting over the weekend. GreyNoise, one of the organizations reporting the in-the-wild exploitation, said in an email that the activity began around 02:17 UTC on Sunday.

"The attacks attempted to download and install one of several pieces of malware on vulnerable devices depending on their specific hardware profile. One such piece of malware is flagged under various names by 40" - four zero - "endpoint protection services. "Security organization Shadowserver has also reported seeing scanning or exploits from multiple IP addresses, but did not provide additional details.

"The vulnerability pair, found in the nas_sharing.cgi programming interface of the vulnerable devices, provide an ideal recipe for remote takeover. The first, tracked as CVE-2024-3272 and carrying a severity rating of 9.8 out of 10, is a backdoor account enabled by credentials hardcoded into the firmware. The second is a command-injection flaw tracked as CVE-2024-3273 with a severity rating of 7.3. It can be remotely activated with a simple HTTP GET request. NetSecFish, the researcher who disclosed the

vulnerabilities, demonstrated how a hacker could remotely commandeer vulnerable devices by sending a simple set of HTTP requests to them."

So thank you and congratulations, Network Security Fish. What a nice public service you have performed for 92,589 D-Link users whom you've just single-handedly turned into victims.

Leo: Yikes. Yikes, yikes, yikes. That's bad news.

Steve: It is. Before long, if not already, those 92,589 D-Link devices will be infected with malware, if they aren't all already. After all, today's Tuesday. And as I mentioned at the start of this adventure, the Internet scan distribution shows that networks in the United States contain more of them than any other single region. I wonder whose networks those devices may be sitting on, and I wonder who might be interested in finding out.

So I suppose the final takeaway lesson for us is that complex devices of this sort that are no longer being actively supported cannot be used safely, at least not in settings such as connected to the Internet, where the security risk is high. Of course, that's easily said; right? It's difficult to retire a perfectly good working device for no obvious reason other than that its manufacturer is no longer active supporting it. A mature policy would ideally rotate such devices into less security-sensitive roles. Give them a place inside the network, behind the firewalls, where they can live out their lives in peace while continuing to be productive. I very much hope that the 92,589 owners of these surviving NAS devices do not experience much hardship as a consequence of Network Security Fish's needless, pointless, and destructive disclosure. It should be clear that there are times when it's far better to just say nothing.

Leo: Wow.

Steve: You know...

Leo: It's like he's showing off, really; right?

Steve: Yes, that is all it - as there's no other justification, Leo. Nobody's learning a lesson. D-Link can't do anything about this. They won't do anything about it. It is, it is just hubris. It is just ego. It is just, hey, look what I did. Look what I found. But he could have said this without, I mean, well, better to say nothing; right? But, you know, he gave a complete, here's how you attack 92,000-plus open, wide-open, innocent devices that are never going to be patched. It's just like, hey, here, take them over.

Leo: I guess...

Steve: Install malware, you know, create a botnet. See whose network they're on. Pivot. Because who knows, some enterprise, some IT guy could have said, hey, I have one of these at home. Works great. Let's buy one for the company. And so it's on the enterprise's network. Someone's going to get in, pivot, now have access to their LAN from this device that is straddling the LAN and the WAN and you know, install ransomware on their network. Why? Because some guy says, hey, look what I found.

Leo: It could have been worse. He could have sold it to Zerodium. Right?

Steve: True.

Leo: I mean, at least he didn't sell it to a nation-state. Not that they would have paid much for it. But still...

Steve: No. Although I'm not sure that this isn't worse because now it's a feeding frenzy.

Leo: It's free, yeah. He's given it away.

Steve: Well, it's every script kiddie who ever, you know, had a wget command, or what's the other one, the famous Linux...

Leo: cURL.

Steve: cURL, cURL, of course. It's like, wow.

Leo: Yeah.

Steve: So Leo, on that happy note...

Leo: Yeah, thanks for cheering me up.

Steve: Why we're here, and we're going to find out why you don't want to use a private note-sharing site.

Leo: Oh, interesting. Yeah, the only D-Link thing I - I used to buy a lot of switches. I think I might have some still rolling around, but those are passive. They're not...

Steve: Yeah, exactly. I think they had really nice-looking little network...

Leo: Yeah, they were nice metal hubs and switches.

Steve: Yup.

Leo: And I think I own a D-Link cable modem. I hope that - that's such a bad practice, to hardwire a backdoor.

Steve: Oh, Leo. It's just...

Leo: It's just depressing.

Steve: I mean, at some point we need legislation where...

Leo: Yeah, you're responsible. Just you're responsible.

Steve: Yes, exactly, exactly. If something happens because of what you did, you are now liable, you know, you're on the receiving side of lawsuits because that's just - you just can't do it.

So get a load of this one. Last Thursday, Brian Krebs of krebsonsecurity.com fame, who really likes doing deep security research, posted a piece that just makes you shake your head. He wrote: "A cybercrook who has been setting up websites that mimic the self-destructing message service Privnote.com" - you know, as in obviously privacy - "accidentally exposed the breadth of their operations when they threatened to sue a software company." Wow. This is chutzpah. "The disclosure revealed a profitable network of phishing sites that behave and look like the real Privnote."

Leo: Oh, no.

Steve: Except that, get this...

Leo: Not so Privnotes.

Steve: Unh-unh. "Any messages containing cryptocurrency addresses will be automatically altered with a different payment address...

Leo: Brilliant.

Steve: It is. It is diabolical. "...controlled by the scammers." So Brian explains, he says: "Launched in 2008, Privnote.com employs technology that encrypts each message so that even Privnote itself cannot read its contents. And it doesn't send or receive messages. Creating a message merely generates a link. When that link is clicked or visited, the service warns that the message will be gone forever after it is read.

"Privnote's ease-of-use and popularity among cryptocurrency enthusiasts has made it a perennial target of phishers" - who could have seen that coming? - "who erect Privnote clones that function more or less as advertised, but also quietly replace their own cryptocurrency payment addresses when a note is created that contains crypto wallet addresses. Last month, a new user on GitHub named Fory66399 lodged a complaint on the 'issues' page for MetaMask, a software cryptocurrency wallet used to interact with the Ethereum blockchain. Fory66399 insisted that their website Privnote.co was being wrongly flagged by MetaMask's 'eth-phishing-detect' list as malicious." Which of course it was.

"Fory66399 wrote," with their arms crossed: "We filed a lawsuit with a lawyer for dishonestly adding a site to the block list, damaging reputation, as well as ignoring the

moderation department and ignoring answers. Provide evidence, or I will demand compensation." So MetaMask's lead product manager Taylor Monahan replied by posting several screenshots of Privnote.co showing that the site did indeed swap out any cryptocurrency addresses.

Leo: Oh. Oh.

Steve: After being told where they could send a copy of their lawsuit, Fory66399 appeared to become flustered. And went silent.

Now, Brian's piece continues with one of his terrific deep dive researches into all the details, and he uncovers a large network of very similar clone websites by backtracking their domain registrations. What I found interesting about this was that this is not hacking some fancy new blockchain technology contract thing that like nobody understands, but to steal like a windfall of $50 million all at once. No. Instead, this is stealing individual small cryptocurrency transactions from cryptocurrency end users. And you can imagine the dialogue; right? You know, "I haven't received the drugs I sent you the money for. What do you mean, you never received the payment? I sent it right after I received the email with your wallet address, and the money was taken from my wallet. If you didn't get it, then where did it go?"

Leo: Who did?

Steve: That's right.

Leo: Maybe Fory got it.

Steve: Uh-huh. Well, gee, it appears likely that it may have made its way into some Russian's pocket, perhaps because you were not paying close attention and used Privnote.co or Privnota.com or Privatemessage.net or Privatenote.io or Tornote.io or Privnode.com or Privnate.com or Prevnote.com. Believe it or not, Brian's research traced each of those Privnote.com malicious copycat domains back to a handful of apparent Russians who had been enjoying fleecing the corrupt Western capitalists.

So we know that it's entirely possible to create a simple website that encrypts a visitor's note on their PC so that it cannot be decrypted except by another third party. The problem is, the technology required to do this is not readily visible and auditable by a site's user. Nor would they understand the crypto code, even if it was visible. The site clearly and cleanly claims that they're unable to read anything that's being sent. And the unwitting user has heard of such sites, like Privnote.com, that's authentic, you know, and that that site arranged to do just that. So it's got a terrific reputation. And Privnote.io, well, it looks the same, so it's probably just the same people who also got that cool .io domain, and it's one fewer characters to type. Might as well use Privnote.io instead of Privnote.com. What could possibly go wrong? You know, and, after all, Privnote.io, it says that it cannot read anything that's sent. So let's just copy and paste our wallet address into it. Right. Good luck with that.

We talked a lot about Zerodium in the past. They're the folks who offer extremely large bounties for new and unknown zero-click vulnerabilities. And unlike the good guys at HackerOne or Zero Day, these creeps sell these zero-days, doubtless at significant profit, to unknown but certainly big-time buyers such as Israel's NSO Group for use by the

Pegasus spyware and almost certainly to governments and intelligence services around the world. In other words, the platform publishers such as Apple and Google are the last to learn of these exploits.

Well now, on Saturday, TechCrunch brings us news of a newcomer named "Crowdfense" - maybe they're fencing, I guess they're fencing the illegal, the ill-gotten goods of a zero-day. And Crowdfense intends to give Zerodium a run for its money. And, you know, this is really not a market where we'd like to see competition flourishing. We'd prefer that the market didn't exist at all.

I have a screenshot in the show notes of Crowdfense's current offering lineup. At the top of the heap they've got SMS and MMS full-chain zero-click compromises, the discovery and disclosure to them of which would net someone selling it somewhere between 7 and 9 million USD. I mean, basically, you find one, and if you check your ethics at the door, you're done for life. Right? I mean, you could probably survive on $9 million. You know, I mean, actually just off the interest.

Okay. So in this case "full-chain" means something that gets the entire job done, not just an "Oh, look, it crashed," but "Oh, look, we now have root access to do whatever we want" sort of thing. A full-chain zero-click for Android brings in 5 million, whereas the same thing for iOS is priced at between 5 and 7. And these are all just within the top paying Mobile Platform category. Crowdfense is also interested in mobile apps, other mobile things, desktop, virtualization, baseband, meaning, you know, the radio that underlies our smartphones, enterprise, web apps, and more.

TechCrunch's headline was "Price of zero-day exploits rises as companies harden products against hackers." Okay, well, that's good. With the subheading "A startup is now offering millions of dollars for tools to hack iPhones, Android devices, WhatsApp, and iMessage." TechCrunch writes: "Tools that allow government hackers to break into iPhones and Android phones, popular software like the Chrome and Safari browsers, and chat apps like WhatsApp and iMessage, are now worth millions of dollars and their price has multiplied in the last few years as these products get harder to hack.

"On Monday, startup Crowdfense published its updated price list for these hacking tools, which are commonly known as 'zero-days' because they rely on unpatched vulnerabilities in software that are unknown to the makers of that software. Companies like Crowdfense and one of its competitors, Zerodium, claim to acquire these zero-days with the goal of reselling them to other organizations, usually government agencies or government contractors, which claim they need the hacking tools to track or spy on criminals." And of course we have lots of evidence we've discussed through the years on this podcast that, you know, politicians and political activists and enemies of powerful people, whoever, end up getting spied on, not just intelligence services tracking known bad guys.

"Crowdfense," they write, "is now offering between 5 and 7 million for zero-days to break into iPhones; 5 million for Android; 3 million and 3.5 for Chrome and Safari zero-days, respectively; and 3 to 5 million for WhatsApp and iMessage zero-days. So the increase in prices comes as companies like Apple, Google, and Microsoft are making it harder to hack their devices and apps, which means their users are better protected."

Okay. So in other words, of course, as zero-days become more rare, they naturally become more valuable. It's good news for everyone that they are becoming more rare. TechCrunch continues: "Dustin Childs, the head of threat awareness at Trend Micro's Zero-Day Initiative said: 'It should be harder year over year to exploit whatever software we're using, whatever devices we're using.' Unlike Crowdfense and Zerodium, ZDI pays researchers to acquire zero-days" - on the other hand, not 7 or $9 million - "then reports them to the companies affected with the goal of getting the vulnerabilities fixed." So those are the good guys, where you get to keep your ethics with you while you accept

some good money, but not enough so that you never have to do anything and can retire on a beach.

"Shane Huntley, the head of Google's Tag Team" - their TAG, Threat Analysis Group - "tracks hackers and the use of zero-days. He said: 'As more zero-day vulnerabilities are discovered by threat intelligence teams like Google's, and platform protections continue to improve, the time and effort required from attackers increases, resulting in an increase in the cost for their findings.'

"In a report last month, Google said that last year in 2023 it saw hackers use a total of" - here it comes - "97 zero-day vulnerabilities in the wild." That was last year, in all of 2023, 97 zero-day vulnerabilities. "And the various spyware vendors, like the NSO Group, which often work with zero-day brokers, were responsible for three quarters of all zero-days targeting Google products and Android. People in and around the zero-day industry agree that the job of exploiting vulnerabilities is getting more difficult.

"David Manouchehri, a security analyst with knowledge of the zero-day market, said that 'hard targets like Google's Pixel and iPhone have been becoming harder to hack every year.' He said: 'I expect the cost to continue to increase significantly over time.'"

"Paolo Stagno, the director of research at Crowdfense" - the new bad guys - "told TechCrunch: 'The mitigations that vendors are implementing are working, and it's leading the whole trade to become much more complicated, much more time-consuming, and so clearly this is then reflected in the price.'"

The first time I read that I thought, trade? What trade? Then I realized that they're calling this zero-day vulnerability finding and selling "a trade." And I suppose it is, though it feels like ransomware gangs talking about their "profit." Profit?

Leo: Yeah. No, no, it's just trade.

Steve: How about theft through extortion? You know, how is that profit? But I suppose that it is, sadly, profitable; though it hardly seems earned. Anyway, the Stagno guy from Crowdfense explained: "In 2015 and/or '16, it was possible for only one researcher to find one or more zero-days and develop them into a full-fledged exploit targeting iPhones or Androids." He says: "Now 'this is almost impossible' as it requires a team of several researchers, which also causes prices to go up. Crowdfense currently offers the highest publicly known prices to-date outside of Russia, where a company called Operation Zero announced last year that they was willing to pay up to 20 million for tools to hack iPhones and Android devices. The prices in Russia, however, may be inflated because of the war in Ukraine and the subsequent sanctions, which could discourage or outright prevent people from dealing with a Russian company. Outside of the public view, it's possible that governments and companies are paying even higher prices.

"David Manouchehri previously worked at Linchpin Labs, a startup that focused on developing and selling zero-days." And again, there's no way that selling zero-days is ethical and cool. Linchpin Labs, unfortunately, was acquired by U.S. defense contractor L3 Technologies, now known as L3Harris, in 2018." And that's encouraging. Anyway, David said: "The prices Crowdfense is offering researchers for individual Chrome Remote Code Execution and Sandbox Escapes are below market rate from what I've seen in the zero-day industry."

Leo: What? What? Wow.

Steve: Wow. A low market rate.

Leo: Holy cow. That seemed like a good price to me.

Steve: Wow. "Alfonso de Gregorio, the founder of Zeronomicon, an Italy-based startup that also acquires zero-days" - so, you know, they're scattered around - "agreed with this, telling TechCrunch that prices could certainly be higher." Well, I guess he wants them to stay low because that's the price he has to pay researchers or hackers who find them. I really don't want to call them "researchers." "Zero-days," TechCrunch writes, "have been used in court-approved law enforcement operations. In 2016, the FBI used a zero-day" - we know where this is going - "provided by a startup called Azimuth to break into the iPhone of one of the shooters who killed 14 people in San Bernardino, according to The Washington Post. In 2020, Motherboard revealed that the FBI with the help of Facebook and an unnamed third-party company used a zero-day to track down a man who was later convicted for harassing and extorting young girls online.

"There have also been several cases where zero-days and spyware have allegedly been used to target human rights dissidents and journalists in Ethiopia, Morocco, Saudi Arabia, and the UAE, among other countries with poor human rights records. There have also been similar cases of alleged abuse in democratic countries like Greece, Mexico, Poland, and Spain. Neither Crowdfense, Zerodium, or Zeronomicon have ever been accused of being involved in similar cases." On the other hand, remember, they're one party removed. They're not the guys who are doing the exploiting of these exploits. They're selling to entities which are then turning around and doing its abuse. So, yeah, these resellers would not be in the loop.

They said: "Zero-day brokers, as well as spyware companies like NSO Group and Hacking Team, have often been criticized for selling their products to unsavory governments. In response, some of them now pledge to respect export controls in an effort to limit potential abuses from their customers.

"Stagno said that Crowdfense follows the embargoes and sanctions imposed by the United States, even if the company is based in the UAE. For example, Stagno said that the company would not sell to Afghanistan, Belarus, Cuba, Iran, Iraq, North Korea, Russia, South Sudan, Sudan, and Syria all on the U.S. sanctions lists."

Leo: That's a really long list. Wow.

Steve: It is. And he said: "Everything the U.S. does, we are on the ball."

Leo: You know why? I bet we're one of their biggest customers.

Steve: Uh...

Leo: Yup.

Steve: Uh-huh.

Leo: We don't want to get them mad.

Steve: I would bet that the NSA is probably taking our taxpayer money and buying these exploits so that they can do things with it. I'll bet you're right, Leo. He said if an existing customer gets on the U.S. sanctions list, Crowdfense would abandon it.

Leo: Mm-hmm.

Steve: "All the companies and governments directly sanctioned by the USA are excluded."

Leo: Of course they are.

Steve: Uh-huh. "At least one company, spyware consortium Intellexa, is on Crowdfense's particular blocklist. Of Intellexa, Stagno said: 'I can't tell you whether it has been a customer of ours and whether it has stopped being one. However, as far as I am concerned, now at this moment Intellexa could not be a customer of ours.'

"In March, the U.S. government announced sanctions against Intellexa's founder Tal Dilian, as well as a business associate of his, the first time the government imposed sanctions on individuals imposed in the spyware industry. Intellexa and its partner company Cytrox is also sanctioned by the U.S., making it harder for the companies, as well as the people running it, to continue doing business.

"Intellexa's spyware has been reported to have been used against U.S. Congressman Michael McCaul, U.S. Senator John Hoeven, and the president of the European Parliament, Roberta Metsola, among others." And finally: "De Gregorio, the founder of Zeronomicon, declined to say who the company sells to. On its site, the company has published a code of business ethics."

Leo: Oh, well. Okay.

Steve: These guys have business ethics, Leo.

Leo: They're so ethical.

Steve: I'm wondering how many sentences or how many words in their ethics statement, "which includes vetting customers with the goal of avoiding doing business 'with entities known for abusing human rights,' and respecting export controls."

Now, reading about these so-called export controls, one has to wonder how difficult it would be for any major country on the U.S. sanctions list to establish a behind-the-scenes relationship with another company in a non-sanctioned region to use as a middleman.

In any event, I thought that checking in on the state of the zero-day market would be useful. While it may not be good news that prices are increasing, since that significantly increases incentives to find the fewer and fewer remaining zero-days that exist, the fact

that prices are rising because these remaining zero-days are becoming ever more scarce, well, that's certainly good news.

I have one bit of miscellany which is happy. Last Thursday, NASA updated the world with the news of the status of the our intrepid Voyager 1 spacecraft. The headline of their posting was "Engineers Pinpoint Cause of Voyager 1 Issue, Are Working on Solution." They explained: "Engineers have confirmed that a small portion of corrupted memory in one of the computers aboard NASA's Voyager 1 has been causing the spacecraft to send unreadable science and engineering data to Earth since last November. Called the Flight Data Subsystem (FDS), the computer is responsible for packaging the probe's science and engineering data before the telemetry modulation unit (TMU) and radio transmitter send the data to Earth.

"In early March, the team issued a 'poke' command to prompt the spacecraft to send back a readout of the FDS memory."

Leo: Is that a Facebook poke command?

Steve: They've poked it. That's similar to a Like, but it dates from 1971, so it's not quite the same.

Leo: Yeah, that's right, the original poke.

Steve: That's right, "which includes the company's software code as well as variables are all being sent back, the computer's software code as well as variables. Using the readout that they received, the team has confirmed that about 3% of the FDS memory has been corrupted, preventing the computer from carrying out normal operations.

Leo: That's fascinating, huh.

Steve: "The team suspects that a single chip responsible for storing part of the affected portion of the FDS memory is not working. Engineers cannot determine with certainty what caused the issue. Two possibilities are that the chip could have been hit by an energetic particle from space, or that it simply may have worn out over 46 years." Just like the rest of us.

Leo: I understand that. I get that. I can understand that completely.

Steve: "Although it may take weeks or months, engineers are optimistic they can find a way for the FDS to operate normally again without the unusable memory hardware, which would enable Voyager 1 to begin returning science and engineering data."

Leo: What a miraculous story. Unbelievable.

Steve: Leo, it is astonishing.

Leo: Isn't it.

Steve: And when you consider, what is it, is it billions or millions of miles away? It is astonishingly far away. How can that thing be pointing at Earth? I mean, talk about, I mean, it's a fraction of a degree at that distance for its antenna, a directional dish, to still be perfectly aligned. To me, that's what is stunning.

Leo: Amazing. And they don't have enough power to send a broad beam. It's got to be a fairly tight beam. So, amazing.

Steve: Right. And, I mean, the power is dropping as we've covered through the years. They're using a radioisotope-based system. Basically, the decaying radioisotopes are heating a thermocouple, which is generating the power to drive this stuff. And they've had over time, as less and less radiation is being produced because it's just, you know, it's just winding down, the power produced has diminished, so they've been having to judiciously turn off successive instruments of their total instrument package because they're looking at the total number of watts being generated. It's just astonishing.

Leo: It is 15 billion miles from Earth,

Steve: That's insane.

Leo: It is incredible.

Steve: Insane.

Leo: Forty-six years, seven months, four days, nine hours, four minutes, and 23 seconds. 15 billion. That is insane. I almost feel like that can't be right.

Steve: I know. And how can it still be pointing with enough accuracy at us? That's mind-blowing to me.

Leo: Twenty-two hours, 22.6 hours light distance away. That's - how did it happen?

Steve: Oh, and there you just scrolled by the instruments which are now on and off on both of the devices.

Leo: Oh, yes, yeah. So one is the first column. Plasma science is off. Imaging science is off. Infrared interferometer spectrometer is off. Quite a bit. But there's still science projects still on, cosmic rays, low-energy charged particles, and magnetometer. Wow. Oh, and plasma web subsystem.

Steve: Yeah.

Leo: Still on.

Steve: And actually we learned that when Voyager 1 passed out of the Heliosphere, the models which cosmologists and astronomers had made were found to be wrong. So it was never expected to be anything other than a fly by some of our planets to take some pictures. But it just wouldn't die. So it kept on going, and it had so much useful instrumentation on it that they're learning new things...

Leo: Incredible.

Steve: ...which are still valuable.

Leo: Patrick said, wait a minute, it's getting closer to the Earth? Yeah, because the Earth's rotating in its direction right now. The distance from the Sun is going up. It's 15.1 billion miles from the sun, or 163 astronomical units.

Steve: So that means that it's moving away from the sun more slowly than the Earth is currently moving around and thus toward it at the moment.

Leo: Right. It's gaining. And then of course as the Earth processes, it will go the other way.

Steve: Right.

Leo: And it'll gain faster.

Steve: And there's only now one radio telescope in Australia which is able to talk to it. And unfortunately it's tasked with doing lots of other things. So they need to like steal a little bit of time at the right moment when they're able to send a burst of instructions to it. And then they wait 44 hours, 22 for it to go out and 22 - I mean, and Leo, the other thing, think of the science we had in '71.

Leo: Oh, yeah.

Steve: That's when I had my first car that had an empty engine compartment because there was an engine and a gas line going to a carburetor. I mean, there was no - and you had a throttle. I had to pull the throttle to start the engine when it was cold.

Leo: Oh, the choke, yeah, yeah, yeah.

Steve: Yeah, the choke, the choke, not the throttle. Right, the choke. And that was the world in '71, were Wozniak was saying, Steve, I think I can get rid of one more Apple computer because, you know, we need the space.

Leo: Look, that's where it is, way the hell out there. Way beyond anything else. Farthest manmade object in the world. In the universe. Unless there's other men elsewhere. But we don't know that, so.

Steve: I don't know where that Tesla coupe is at the moment.

Leo: A lot closer, I can promise you.

Steve: I think so.

Leo: Wow. Just a great story. It's such a great story.

Steve: And again, 1971 technology, it's just astonishing.

Leo: But think, I mean, in 1969 we landed on the moon, two years before that. And we're having a hard, a devil of a time doing it again. So maybe those guys back in the '60s knew something.

Steve: Yeah, the bits were bigger, and so they were more robust back then. What you want is bigger bits, really.

Leo: Yeah, our bits have gotten way too small.

Steve: Speaking of bits, I was tempted to name this podcast "SpinRite 6.1" because what happened Sunday afternoon means so much to me. But since it doesn't mean that much to the rest of our listeners, that didn't seem appropriate. What happened on Sunday is that I finally updated GRC for the first time ever to begin offering 6.1 as its official SpinRite. So 6.1 is finally what new purchasers will receive when they go to GRC for the first time. So, you know, it's huge for me. I've been living with my commitment to offer - and I don't have the website all...

Leo: Oh, yes. So it's still 6.0.

Steve: So if you go to the top, Leo - I know, I've just got this done. If you click on the menu under SpinRite and the top left...

Leo: Okay.

Steve: There's a little - yup.

Leo: Upgrade to 6.1, okay.

Steve: Yup. Or if you just click on Purchase SpinRite, go down a few, and then you'll see...

Leo: There it is. Woohoo!

Steve: ...that it is 6.1 is what we are offering.

Leo: That's pretty cool, Steve. By the way, this website looks like it came from 1971. Congratulations.

Steve: I came from 1971.

Leo: But the bits are bigger here. They really are.

Steve: That's right, the bits are bigger, baby.

Leo: Is the website written in assembly? Tell the truth.

Steve: It's all hand-coded. I wrote it in HTML.

Leo: Nice. Nice.

Steve: Before CSS even.

Leo: Yeah, yeah, yeah.

Steve: And I do look at the code, and it hurts, actually. And I've had a number of our listeners who've said, Steve, Steve, Steve.

Leo: No, no, no.

Steve: I'm a website designer. Let me fix this for free. I will be happy to fix it for free.

Leo: No, don't. Okay.

Steve: And I, you know, I want to do it. One of the problems I had with employees was they were having all the fun doing the work. And so consequently...

Leo: Yeah, but you update that, and then you've got to maintain it. Trust me, as somebody with a modern website, stick with what you've got. Static is a good thing.

Steve: I am. Anyway, I've been living with my commitment to offer 6.1...

Leo: Yay.

Steve: ...for so long now, you know, it's been more than a decade, and I felt guilty whenever I'd stolen time from that. You know, and now every time I remember that 6.1 is there being sold, it's like, oh. I mean, it gives me a great sense of peace.

Leo: Good for you.

Steve: One thing I wanted to mention, though, to our listeners is that, after sharing the experience my wife and I had with her Dell laptop during last week's podcast, I did notice an uptick in SpinRite sales; in other words, more "Yabba Dabba Doo's." Now, I'm fine with that since many people have reported significantly improved performance after running SpinRite at Level 3 over an SSD, which is almost certainly going to also improve the long-term reliability of their system. So I can stand behind the benefits that people are likely to see. My only concern was that until the afternoon of this past Sunday, April 7th, which also happened to be my third wedding anniversary, so it was a busy day...

Leo: Oh, congratulations.

Steve: Unless those new purchasers knew to then go to GRC's prerelease.htm page, they would have obtained 6.0. So I just wanted to make sure that anyone who may have heard that and been excited and went out to get SpinRite, that they knew that they were getting 6.0, and everyone should go get 6.1. You can use your receipt that you received to bring up the page and download, you know, 6.1 is what we're offering now. So anyone who goes to get it now who bought 6.0 will get 6.1, or you can go to the /upgrade.html. Greg said to me, my tech support guy, why are we still calling it "prelease"? And this was, I think, yesterday. I said, oh, yeah. So I created another page called upgrade.htm, so you can use that, too.

Anyway, the second thing I wanted to share was inspired by someone using the handle "The Big Bear" who posted over in GRC's SpinRite newsgroup. After running SpinRite 6.1 on two Macs, he posted: "And now I have two Macs tested and refreshed." He said: "And it makes quite a difference. The frequency with which the colorful beach ball had shown was starting to worry me, and it has all but disappeared now. And the startup times feel halved, at least." He said: "Wish I had measured it before and after." He said: "Documentation is not my strong suit, but I will put it on my to-do list, to write an update mentioning the extra hoops required with the latest Sonoma macOS."

Now, the reason I'm mentioning that, aside from it being another instance of welcome feedback about my use of the past 3.5 years creating 6.1, is that while FreeDOS and SpinRite will run on Intel-based Macs, getting them to boot from a CD or USB can be a bit tricky. So I wanted to remind any would-be Mac purchasers that this is the reason I created GRC's freeware named "Bootable," in favor of "DOS Boot," although that was, you know, I was tempted to name it that. You know, you can get Bootable. Download it for free. If you can get it to congratulate you on your success in booting it, then exactly the same path can be taken with SpinRite.

And as I said, Bootable can be freely downloaded at any time. And GRC's web forums at forums.grc.com contain a growing knowledge base of help for Mac users. And that's where this Big Bear guy will post instructions and more details, although there's already a lot of stuff there because, you know, Mac users have said, hey, how do I get this to go? So anyway, next up is email. I know our listeners can't wait to for GRC to have a mailbag that they can send things to. And then is to update the documentation. And then this project will be finished. So yay.

Leo: And now what? Next, after this?

Steve: I know what, but...

Leo: Oh, it's a secret.

Steve: ...I'm not going to say yet.

Leo: Okay. Smart. Wise.

Steve: One thing at a time. I always get myself in trouble by overcommitting.

Leo: Exactly.

Steve: And precommitting. And my wife, bless her heart, says, "Now, you're not going to promise anything again, are you?" I was like, no, I'm still recovering from the last promise.

Leo: Yeah. Well, she's put up with this, if you got married three years ago, for the entire life of your marriage. Maybe it's time to take a little break.

Steve: No, she is a dream.

Leo: Yeah, that's awesome. All right. Minimum viable - I know what a minimum viable product is. When you're an app developer or creating a new site, you create the minimum viable product. You know? But what is this MVSP?

Steve: Yup. And I'm sure that's where they came from. So our beloved industry is slowly, very slowly, getting its act together. But it is happening. And I've been encouraged by some recent news surrounding the Minimum Viable Secure Product effort. The group's list of contributors has been growing, and it now includes some well-known names such as Salesforce, Google, Okta, Slack, Vanta and about 20 others. I guess that's Okta. The reason this is today's primary podcast topic - aside from the whole thing being an extremely worthwhile effort, which it is - is due to the announcement of the effort's latest member, which was made on Thursday last week.

The posting reads: "Today, we're excited to announce that CISA is joining the Minimum Viable Secure Product Working Group. Since launching CISA's global Secure by Design initiative last year, we've received a tremendous amount of feedback, including through our Request for Information that recently closed." And of course it was the Secure by Design initiative where we first saw this notion suggested of requiring a manual action on the device in order to change the configuration in any way that could be foreseen as being dangerous, which I think is brilliant because, you know, again, inconvenient, yes, but it's the right thing to ask for.

Anyway, so CISA's been leading on a lot of this. They said: "One of the key questions we've gotten is how organizations consuming software can ask the right questions of their software manufacturers. Such a 'secure by demand' approach" - as opposed to secure by design, which is what CISA's is - "is crucial to drive the uptake of secure by design principles and practices. Too often, procurement questionnaires are filled with long lists of questions which don't always correlate with positive security outcomes. In order to achieve a future where technology is secure by design, companies buying software should have simple and to the point questions for their vendors.

"The MVSP is an important step toward this goal. MVSP offers a simple checklist that organizations can use to strengthen security at multiple stages - to review their software vendors' security during procurement, as a self-assessment tool for their own software, as part of their software development lifecycle, or as contractual controls - which can go a long way toward helping ensure secure by design principles are followed. We're excited to join the MVSP working group to help shape the direction of the initiative going forward. The MVSP is composed of a broad coalition of technology manufacturers, and the working group is open for anyone to join."

Okay, now, reading through the MVSP's checklist put a smile on my face, as I mentioned at the top of the show, as I imagine it will for our listeners, since we have carefully examined many of these issues in the past here. What everyone is hoping is that powerful technology procurement bodies like for example the various branches of the U.S. government, including both its civilian and military bureaucracies might start making an adherence to these principles more than just requests, but make them mandatory for future purchases.

The MVSP web site is just MVSP.dev, and they explain their mission by writing: "Minimum Viable Secure Product is a list of essential application security controls that should be implemented in enterprise-ready products and services. The controls are designed to be simple to implement and provide a good foundation for building secure and resilient systems and services. MVSP is based on the experience of contributors in enterprise application security and has been built with contributions from a range of companies.

"We recommend that all companies building enterprise software" - and notice today, today it's a recommendation. Let's see how this evolves over time. It would be just terrific if it became more than a recommendation. So "We recommend that all companies building enterprise software and services, or otherwise handling sensitive information, implement the MVSP controls and, where possible, go well beyond them in their application security programs. We welcome constructive feedback to help us continue to improve MVSP and provide a control set that meets the needs of its users."

Okay. So MVSP is a list of essential application security controls at the enterprise level. Let's look at what they are. So under business controls they have: "Publish a vulnerability disclosure policy that outlines the testing scope, provides a legal safe harbor, and gives contact details for security reports." Okay. Those are all things we've discussed in this podcast. Vulnerability researchers should be free, should know that they're free to research a company's products, the security of them, while being legally

protected from retribution or reprisals if they hack a supplier's product without malicious intent for the sole purpose of discovering and responsibly reporting vulnerabilities, even if it's for pay. You know, if it's to report them to HackerOne or a legitimate service. But the point is, publish a vulnerability discovery and disclosure policy and make it clear.

Also, flesh out, or rather fleshing that out, the MVSP lays out the required components: "Develop and document procedures for triaging and remediating reported vulnerabilities, respond to reports within a reasonable time frame, and patch vulnerabilities quickly." They also suggest contracting with a security vendor to perform comprehensive penetration testing of products, services, and dependent systems at least once a year. We know that's been done, but we know that's not probably common practice enough. So yes, you need third-party eyes on something. Your own developers cannot aggressively test the security of the products they develop. It just doesn't work.

They have: "Notify relevant parties about any security breach that affects sensitive information no later than 72 hours upon discovery, and upon learning any additional details of the breach. Consider reporting the breach to relevant national cybersecurity agencies in line with local guidance and regulations. In any such reporting, include the nature of the breach, relevant contact information, the consequences of the breach, and the measures taken and needing to be taken to remediate the issue."

"Be certain to sanitize all storage media holding unencrypted production data. Implement single sign-on using modern, maintained, and industry-standard protocols for all customers at no additional cost. Redirect traffic from HTTP (port 80) to HTTPS (port 443). Exceptions to this are internally secure protocols designed to run over unencrypted connections," you know, such as OCSP, the Online Certificate Status Protocol, that doesn't need it. And I also noted, whenever I've been doing digital signing of code, the connection to the timestamp server is just HTTP because it provides - it's another example of a protocol that uses its own internal security.

Also, "Include Strict-Transport-Security header with a long max-age value and set authentication cookies as Secure." We know that this prevents the browser from ever sending those cookies out over non-encrypted connections. They've got "Apply appropriate HTTP security headers to reduce the application's attack surface and limit post exploitation. These should include setting a minimally permissive Content Security Policy," you know, the CSP, "limiting the ability to in-line frame the application by enabling framing controls with X-Frame-Options or CSP frame-ancestors and disable caching for APIs and endpoints that return sensitive data."

Again, these are all, like, these are all things that web designers should do. But they don't unless they have to, or unless they're told to, or unless there's some policy to make that happen. I remember doing that for the SQRL queries and responses from the SQRL server. And, you know, it's a little nerve-wracking because you're - especially nerve-wracking to add them after the fact because you're not sure what you're going to break. If maximally restrictive policies are there from the start, when you add a feature and test it, and it doesn't work, then you can look at how to make the smallest change required to loosen the policy to allow the technology you want to work to work. That's entirely different from just not having anything at all, where everything works. The problem is lots of things you don't want to have work will then also work.

And of course the MVSP also had a lot to say about password policies. They wrote: "If password authentication is used in addition to single sign-on, then" - and we have a series of bullet points: "Do not limit the permitted characters that can be used." Yay. "Do not limit the length of the password to anything below 64 characters. Do not use secret questions as a sole password reset requirement. Require email verification of a password change request, and require the current password in addition to the new password during password change. Store passwords in a hashed and salted format using a memory-hard

or CPU-hard one-way hash function. Enforce appropriate account lockout and brute-force protection on account access. And do not provide default passwords for users or administrators."

And what about the use of security-sensitive third-party libraries? You know, like Log4j. They say: "Use modern, maintained, and industry-standard frameworks, template languages, or libraries that systemically address implementation weaknesses by escaping the outputs and sanitizing the inputs. Ensure third-party dependencies are maintained and up-to-date, with security-relevant updates having a security score of medium or higher applied in line with your application patching schedule. Upon becoming aware of a Known Exploited Vulnerability" - that's CISA's KEV collection - affecting a third-party dependency, the patch should be prioritized. Where dependency patching or upgrades are not possible, equivalent mitigation should be implemented for all components of the application stack."

And so we can sort of group all of this as things that somebody trained in modern security measures would know, but also things that are, you know, not fun to spend time on; right? It's like, gee, what did you do all last month? Well, I brought us more into compliance. What? Well, you know, what new functions work as a result? Uh, well, none. But we're more in compliance than we were. The point is, you know, it's thankless; right? It's not until everybody around you gets attacked and hacked, and you don't, that you begin to look like a star.

So anyway, hardly surprising, these guys are also big fans of logging, recommending that logs be kept of all authentication events, both successes and failures. And I thought that was interesting; you know? Log all security-relevant configuration changes, including of course disabling of logging, and log application owner access to customer data to provide access transparency. The logs must include the user ID who's involved, the IP address from which anything is happening, a valid timestamp, the type of action performed, and the object of the action. Logs must be stored for at least 30 days at no additional charge - I thought this was interesting - to the client or customer. We know that we've recently seen instances where some cloud provider said, well - actually Microsoft in particular - we'll provide you with more advanced security logging, but that's an extra cost option. And as we know, they've backpedaled on that a bit. And not surprising that the logs should not contain sensitive data or payloads.

And though it barely needs saying, they recommend using current, maintained, industry-standard means of encryption to protect sensitive data in transit between systems, and at rest in all online data storage and backups. And when vulnerabilities are found, they say: "Produce and deploy patches to address application vulnerabilities that materially impact security within 90 days of discovery. For vulnerabilities with evidence of active exploitation, production and deployment of patches should be prioritized." Okay, that one is a "duh." And finally: "Publish a security bulletin that details the vulnerability and its root cause if the remedy requires action from customers. We've seen many instances where permissions were far too open. We've noted that nothing breaks when everyone can access everything, so it's often discovered after a breach of some kind that the source of the breach was overly permissive access controls.

So they write: "Limit sensitive data access exclusively to users with a legitimate need. The data owner must authorize such access. Deactivate redundant accounts and expired access grants in a timely manner. Perform regular reviews of access to validate need to know. Ensure that remote access to customer data or production systems requires the use of multifactor authentication."

And I know, yeah, everyone knows, all of that is obvious. I will say, you know, I'm subjected to it by Level 3 because my badge, they insist on continually expiring my badge. And it's annoying because it's me. And I have to make sure that I'm keeping it

renewed because, if it does expire, then renewing it is a bigger pain. And if I need to run over there, I need to have my badge current. And they're constantly expiring it. On the other hand, yes, it's more secure. If it were not just me, if it were an organization where, for example, 20 people had access, the act of having to renew all of those badges would be like, oh, wait a minute, Herman no longer works here. So it's, yeah, I'm not renewing that badge. But if it didn't expire, your attention wouldn't be brought back to it. So, you know, all of these things, they're little incremental increases in pain, but they're important because they do things.

So, you know, nobody wants to sit and make time to review access permissions. It's boring. And, you know, it might not even be productive. You may not find anything that needs to be changed, while a million other actually important things need to be done. As a consequence, typically, it never happens. But it can be a big source of problems. So they also advise that it's important to maintain a list of third-party companies with access to customer data, which can and will be made available to clients and business partners upon request.

And what occurred to me is since that list, providing that list, might be embarrassing if, for example, it were to contain the names of contractors who were no longer affiliated with the organization, this also, again, forces a review and provides some incentive to remove access once relationships have terminated, and instances where the access doesn't, you know, remove itself. And again, through the years we've seen that instances where that not happening has come back to bite companies. And since it often doesn't happen automatically, it's another of those things that often falls through a crack.

They close this comprehensive list, you know, of things that, as I said, everybody knows would be good to do, but many organizations are still not doing, by reminding about the need for and importance of backups. They recommend not only securely backing up all data to a different location - you know, Leo, this is your standard, you know, 3-3-3 or whatever it is, backup system.

Leo: 3-2-1, yes.

Steve: 3-2-1, right. I knew there were some numbers involved. And, you know, we all know that everybody would like to be in full compliance with these guidelines. And we also know about inertia, and that few things change on their own for the better, or for that matter change at all. It is for this reason that these MVSP guidelines exist, and it's the reason CISA has added their name to the group's growing list of contributors.

The bottom line is that doing all of these things would come at some cost, and most businesses are looking for ways to cut costs, rather than ways to incur additional expense. And the businesses don't pay the price until they get bit by a security problem. So it's going to be a difficult lift. But at least now all of these useful concepts have been pulled together in a single place, MVSP.dev. And if at some point world governments were to require compliance, well, then everyone who wanted to sell to those major markets would need to clean things up. And that would help everybody.

Leo: Well, it's a step.

Steve: Yes. It is. And, you know, we're not going to get there without it.

Leo: Right.

Steve: It doesn't mean we're going to get there with it. But, you know, it's better to have it than not.

Leo: Right.

Steve: Bye.