The Maker's Schedule

Description: This week we examine the collapse of one of the four NIST-approved post-quantum crypto algorithms. We look at what VirusTotal has to tell us about what the malware miscreants have been up to, and at the conditions under which Windows 11 was corrupting its users' encrypted data. We also celebrate a terrific-looking new commercial service being offered by Microsoft, and we briefly tease next week's probable topic, which is cryptographer Daniel Bernstein's second lawsuit against the United States. I want to share a bunch of interesting feedback in Q&A style from our terrific listeners, then I want to share my discovery of a coder, serial entrepreneur, and writer by sharing something he wrote which I suspect will resonate profoundly with every one of our listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-883.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-883-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Yes, it's a Patch Tuesday, but we'll save deets on that till next week. Just, you know, we just like to prepare you for what is about to come. We are talking about a whole bunch of interesting things. Steve's found a new favorite writer, Paul Graham. It's not sci-fi; it's fact. We'll talk about that. We're also going to talk about the failure of one of NIST's proposed solutions for post-quantum crypto. And deception at scale. What's the biggest problem with viruses today? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 883, recorded Tuesday, August 9th, 2022: The Maker's Schedule.

It's time for Security Now!, the show where we cover the latest in security, privacy, bad guys, good guys, white hats, black hats, gray hats, and Steve Gibson, who doesn't need a hat.

Steve Gibson: And even really smart people.

Leo: Even them.

Steve: Even them. We've got two really smart people that are topics: Dan Bernstein, who is one of my favorite cryptographer mathematicians, and the person who's responsible for the show's topic, a guy by the name of Paul Graham, who I'll be introducing people to who may not know of him. He really never crossed my radar one way or the other, although he's long been on yours, Leo.

Leo: Oh, yeah.

Steve: So this is Security Now! Episode 883 for the 9th of August. This is Patch Tuesday.

Leo: Whoo, boy.

Steve: So next Tuesday we'll find out what happened.

Leo: Now I know why my Windows machine was so slow starting up this morning. Ahhh.

Steve: Yeah, and why I've been saying no, no, no, please don't reboot right now, I'm trying to do a podcast.

Leo: Yeah.

Steve: So we're going to examine the collapse of one of the four NIST-approved post-quantum crypto algorithms.

Leo: Oh, good, I was hoping you'd do this, yes.

Steve: We just announced them a couple weeks ago, and remember Scotty's Dilithium crystals and so forth. Anyway, we lost one. But the lessons it teaches us are very important. We're also going to look at what VirusTotal has to tell us about what the malware miscreants have been up to lately, based on all of their stats. And at the conditions under which Windows 11 was corrupting its users' encrypted data. Yes, got to love those upgrades. We also celebrate a terrific-looking new commercial service being offered by Microsoft. I'm not quite sure what it is from their description, but it looks great. And we briefly tease what will probably be next week's topic, which is cryptographer Daniel Bernstein, as I mentioned, his second lawsuit that he's filed against the United States.

So before we get into the other stuff, the end of the show stuff, we'll talk about that. I also want to share a bunch of interesting feedback, sort of more of in a sort of a Q&A style, from our terrific listeners. And then we're going to wrap by - because I want to share my discovery of a coder, a serial successful entrepreneur, and a surprisingly good writer by sharing something that he wrote which I suspect will resonate profoundly with everyone of our listeners. So today's title is part of the title from one of Paul Graham's essays. This is "The Maker's Schedule." And, oh, we've got a great Picture of the Week. I will do my best to describe it. I saw it, and I thought, oh, this is just too perfect.

Leo: Oh, a big show coming up. And I'm looking forward to, yeah, Paul Graham has been around for ages. He founded Y Combinator, which is a big startup incubator in the Bay Area, and very well known.

Steve: More than 3,000 startups. And, boy.

Leo: Yeah, some of them huge.

Steve: And many that we all know.

Leo: Yeah, absolutely. But also a great thinker, I agree with you, brilliant writer and essayist and Lisp aficionado.

Steve: Yes, he is, near and dear to your heart.

Leo: You bet. All right. Back to Steve and the Picture of the Week.

Steve: Okay. So the way to describe this. We don't often get, like, thick fog in Southern California. But I spent the first third of my life in Northern California.

Leo: Yeah, we get it, mm-hmm.

Steve: Oh, boy. And especially like Sacramento, driving around the Sacramento area, for some reason like this super thick fog. So remember how, if you've ever driven in fog, how all you can see are the lights of an oncoming car. It's just two white glowing things in the darkness. Or even in the day, depending upon what time of the day the fog is. But so the point is that the fog obscures everything that's not lit up. Okay.

So, and I have to say, this Picture of the Week, it's worth getting the show notes just to see this because this was an actual photo taken. In the distance is a digital billboard which has - I know, it's just so good - which has crashed or for some reason caused its backing Windows OS to display an error message in a dialog box. So you don't see the normal - so the billboard itself is not displaying whatever ad it normally would. Instead, it's this dialog box. And because this is taken on a foggy night, all you see, like, hovering in the sky is this eerie glowing Windows error dialog. And so the caption on this great photo says "A digital billboard in Odessa malfunctioned, in the fog, convincing unknown numbers of motorists not only were they living in the Matrix, but it was being run on Windows 98." So anyway...

Leo: That we know can't possibly be true.

Steve: Yeah, I don't think that's the case. But anyway, just a great picture. So thank you, whomever it was, a listener of ours who sent that saying I think this would be a Picture of the Week. Boy, were you right. Absolutely.

Okay. So we already know crypto is hard. In fact, it's even harder than we know. Okay. How many times have we observed the fallacy of rolling one's own crypto? And even when professional academics carefully design, test, and vet an algorithm, they sometimes get it wrong. The biggest news of the past week in the crypto sphere is the fall of one of those four final and chosen post-quantum cryptographic algorithms. But to their credit, even while NIST was announcing their final status, recall that everyone was told not to commit them to code just yet. Well, that turned out to be sage and prescient advice.

Some guys whose work we've covered in the past, the researchers with the Computer Security and Industrial Cryptography group at KU Leuven managed to break one of those four late-stage candidate algorithms which was ready to be deployed, like soon, for post-quantum encryption. Fortunately, the Dilithium crystals are still intact. Those algorithms are okay. The algorithm they cracked, SIKE (S-I-K-E), which is the abbreviation for Supersingular Isogeny Key Encapsulation, it made it through all of the earlier stages of the U.S. Department of Commerce's National Institute of Standards and Technology, the NIST competition.

So that competition has been running since 2017, five years ago, and started out with 69, 69 candidates in its first round. That weeded the number down two years later to 26 surviving candidates from the 69. That is to say, all the other ones, something was, like, problems were found. It could have been performance. Most likely it was some security problems, like there were defects found in some of those. Then the next year, which brings us to 2020, the third round, we had seven finalists with eight alternates. And as we know, when we talked about this a couple weeks ago, this was the end of the fourth round where we had three finalists and one alternate being selected to become the standards.

So the good news is, even after the announcement of the final four, or perhaps because of the announcement of the final four, the KU Leuven researchers rolled up their sleeves and got to work. They approached the problem, however, from a different angle than the cryptographers who designed it. They came at it from a pure math angle, attacking the core of the algorithm's design instead of any potential code vulnerabilities, which is actually how a lot of these previous candidates got weeded out, as I mentioned.

SIKE's base encryption was based upon something known as Supersingular Isogeny Diffie-Hellman, or SIDH for short. But the mathematicians were able to show that SIDH itself was in turn vulnerable to a pure math attack which had been developed back in 1997 known as the "glue-and-split" theorem, which is an attack on elliptic curves. In response to this, like the discovery of this, SIKE's co-inventor, David Jao, a professor at the University of Waterloo said: "The newly uncovered weakness is clearly a major blow to SIKE." Which is, you know, you can imagine he's feeling a little protective of his algorithm. So in other words, it killed it. Or as McCoy would have said, "It's dead, Jim." And he added that all this goes back to cryptographers' sometimes imperfect dominion of pure mathematics, in that the approach taken by the researchers, he said, was "really unexpected." Uh-huh. Well, in any event, unexpected or not, it wasn't going to be good enough.

So essentially this means that the researchers used math rather than mechanics to understand SIKE's encryption scheme and were then able to predict and then retrieve its encryption keys, which is specifically what it was designed to protect. And for their efforts they received a bounty award of $50,000 from Microsoft after they published their paper titled "An Efficient Key Recovery Attack on SIDH." And although not directly on SIKE, SIDH being the basis for SIKE, they are the same.

The people who've been watching and participating in this process are concerned that this appears to be as difficult as it is. In a note which was written, an email actually, to Ars Technica which Ars Technica published, Jonathan Katz, an IEEE member and professor in the Department of Computer Science at the University of Maryland, wrote: "It is perhaps a bit concerning that this is the second example in the past six months of a scheme that made it all the way to the third round of the NIST review process, and in this case the fourth round, before being completely broken using a classical algorithm."

He added that: "Three of the four post-quantum crypto schemes rely on relatively new assumptions whose exact difficulty is not well understood, so what the latest attack indicates is that we perhaps still need to be cautious and conservative with the

standardization process going forward." In other words, we knew crypto was hard, but now it appears that it might be even harder than we thought it was. And part of the process is we're stepping into it like into unfamiliar territory, pre- and post-quantum. They're, like, it's a whole different thing, a whole different nature of computing that we're needing to be hardened against compared to classical computing, which is what we've been using until now. Right? So there's a lot of understanding and basis for grokking the way computers work today. That's not the way they work tomorrow.

So we've already been doing this work, working to come up with the algorithms which we'll be able to rely on post-quantum, for five years. And thank goodness that as an industry we started as early as we did, and that we're now focusing upon this as intently as we are. Even though it's true, and you and I, Leo, have had some fun with this, that sufficiently powerful quantum computers still seem to be a safe distance away - have they factored the number 31 yet? I don't know - it's increasingly looking...

Leo: I think 31's a prime, though, so I think they may be a way off. If they factor 31, then we've got a problem.

Steve: Okay. How about 27? Let's go with that.

Leo: There you go.

Steve: Or 35 I think is what I meant. Anyway, it's interestingly looking as though we're going to need more time than we thought to get ready. And remember that everything being encrypted and stored today, still under pre-quantum crypto, is potentially vulnerable to disclosure once capable, sufficiently capable quantum computing emerges. And at some point, depending upon what it is that's being stored, it might even be worth upgrading existing stored encrypted data at rest from its current pre-quantum encryption to post-quantum encryption. In other words, read it, decrypt it under pre-quantum, re-encrypt it under post-quantum, and then store it back.

Now, of course the NSA, they're happily storing everything they can get their hands on, presumably, that is pre-quantum. And I wouldn't be surprised if they're one of the first people to order one of these big monster post-quantum machines, as soon as they are available.

Leo: So they have, so just to be clear, it's kind of confusing. You mentioned Dilithium, Crystals Dilithium. They have four, already standardized four candidate algorithms which are Crystals-Kyber, Crystals-Dilithium - Crystals-Kyber is the KEM or Key Establishment Mechanism. And then there's Crystals-Dilithium, Falcon, and SPHINCS. This was part of a second class of four that - SIKE was part of a second class of four that they were looking at.

Steve: Right, right.

Leo: So it doesn't undermine what they've already done, although it does, as you say, make us wonder, you know, how good these things are. And I think they've already - they already say they're going to standardize on Crystals-Falcon and SPHINCS; right?

Steve: Correct. So the lattice-based algorithms, cryptographers and mathematicians have been working with lattice-based crypto already for some time. So it's reasonable to hope that those are going to be stronger and more resistant to I guess what we would call "new approaches" to cracking them. And of course we want every possible approach; right? I mean, we want something that is absolutely resistant because the worst thing that could happen would be that the whole world standardizes on something that we all collectively believe is super solid, and then some complete crack is found.

Leo: Whoops. You mean like DES or something like that.

Steve: Kinda like that.

Leo: Yeah.

Steve: Although generally we've, you know, in the case of DES it was the NSA who said, oh, you know, I think maybe you want to do that three times, rather than just once. And thus triple DES was found to be sufficiently secure because just doing it three times, each time with a different third of the whole key resulted in enough security margin. So the nice...

Leo: So this does point out this is a good process.

Steve: Yes.

Leo: That's what you hope is that somebody comes along and says, no, you can't use that one. That's why you do this.

Steve: Oh, yes. And so this was useful in two ways. It both took one of the not-safe-enough contenders out of consideration, and it sort of shocked everyone. It's like, wait a minute. And this is the point that this guy was making was this thing made it through five years of scrutiny, down to the fourth round, where people were so tired they finally said, okay, fine. We've been picking away at this thing for five years. There are these four. They all look fine. Well, they were wrong about one of them. Are we sure they're not wrong about the other three?

Leo: Right, right. Well, that's the point. They weren't 100% sure of any of those four. They aren't the four candidate algorithms that they've put forward for standardization. These were the second choices anyway. But I'm glad they got rid of one.

Steve: Got you.

Leo: And I hope they continue to look at the four that they have decided to standardize.

Steve: And it's fair to say that they got rid of the one that was less well understood of the four.

Leo: Yeah. They liked it because it had a small key.

Steve: Yeah. And I like the other ones because they've got Dilithium in them.

Leo: Can't go wrong with that.

Steve: That's great; you know? Either warp engines or light sabers. One way or the other. Okay. So VirusTotal. Last week they published a report titled "Deception at Scale," where they laid out the terrain of the malware samples that are uploaded to them more or less constantly to be analyzed. They sit in the perfect place to see what's going on. They've got great scope.

I've explained in the past that signing my own executables, I've discovered the hard way, because people were saying, hey, Windows is saying this is not safe, you've got a virus, it's like, no, I don't. Actually, it didn't say that. It just said this is, you know, you don't have any reputation here. So the point is that signing my executables was not sufficient proof of the integrity of my apps to bypass various of what are now hair-triggered malware cautions.

But VirusTotal reported among other things, get this, that fully 87% of the more than one million malicious samples which were signed at the time they were uploaded to VirusTotal since the start of last year, January 2021, contained a valid signature. 87% had a valid signature, those that were signed. So what that tells us is that signing code no longer means much. It's necessary, but not sufficient. The bad guys are arranging to obtain code-signing credentials, just like any other legitimate code publisher would. Just like I do.

So moving forward, the only thing that can be used, that is, can be relied upon, is the reputation of the hash of a given executable that is earned over time. Any new hash will need to start over from scratch earning the reputation that that specific exact code that it's the hash of is trustworthy.

And there was another little interesting tidbit. If you care to protect yourself somewhat by inspecting the Certificate Authority who issued the Authenticode certificate that was used to sign a program which you're considering running, it's worth noting that more than half, actually more than 58% of the most-often-abused code-signing certificates were all issued by just one company, a Certificate Authority known as Sectigo. And if the name Sectigo isn't ringing any bells, it's probably because they renamed themselves after their repeated conduct spoiled and soiled their previous name, which was Comodo. We've talked about Comodo quite a bit in the past, all the different mistakes they made like allowing people to create their own certificates through problems in their web interface and giving certificate minting authentication to people who didn't warrant it and so forth.

Anyway, I imagine that they're the favorite of malware authors mostly because their certs are less expensive than the competition. And really it's not their fault that VirusTotal sees most malware signed by their certs, since anyone can purchase a code-signing certificate from any certificate authority, so going to go with the cheapest. I don't, but I don't want to be signed by Comodo, now named Sectigo. And the whole thing is roughly analogous to what Let's Encrypt did to TLS connections; right? Once upon a

time having a web server certificate meant something. Not anymore. Today, everyone needs to have one, and they mean nothing because they're just being minted by automation based on the domain of the server that they're sitting behind. So okay.

Anyway, VirusTotal also revealed that the top three most-often-spoofed programs were Skype, Adobe Reader, and VLC Player. Malware is masquerading as those three utilities - one of those three, Skype, Adobe Reader, and VLC as the top three - as basically, obviously, as a means to abuse the well-earned trust that they've earned, that those apps have earned with users everywhere. And while those are the top three, the top 10 are rounded out by 7-Zip, TeamViewer, CCleaner, Edge, Steam, Zoom, and WhatsApp. So, yeah, the top of the popular apps that people are needing now to grab wherever they are.

So VirusTotal said in their report last week: "One of the simplest social engineering tricks we've seen involves making malware look like a legitimate program. The icon of these programs is a critical feature used to convince victims that these programs are legitimate." Just the icon. Of course, no one is surprised that threat actors employ a variety of approaches to compromise endpoints by tricking unwitting users into downloading and running seemingly trusted executables.

The other way this is achieved is by taking advantage of genuine domains, at least the top-level or second-level domains, to get around IP-based firewall defenses. Some of the most abused domains which VirusTotal has seen are discordapp.com, squarespace.com, amazonaws.com, mediafire.com, and qq.com. In total, more than 2.5 million suspicious files were downloaded from 101 domains belonging to Alexa's top 1,000 websites. In other words, 10% of the top 100 website domains have been used as sources for malware. And the misuse of Discord has been well-documented, with that platform's content delivery network becoming a fertile ground for hosting malware alongside Telegram, while also offering a perfect communications hub for attackers.

So ultimately, checking anything that's downloaded which might be suspicious against VirusTotal, I think, is the best thing anyone can do. As I mentioned a while ago, back when I was needing to bring old DOS machines onto my network in order to debug SpinRite on them, I was sometimes needing to go to well-off-the-beaten-path driver repositories to locate old drivers for old network adapters. Driver repositories are classic sources of malware.

So in every case, I ran anything that I downloaded past VirusTotal to make sure that it didn't raise any alarms. And normally you get like one or two, some weird obscure, you know, VirusTotal I think scans across or against as many as 75 different virus, you know, antivirus engines. And you'll typically get a couple reds, misfires, false positives from some scanners you've probably never heard of. And so that's not a problem. It's when you see like 20 or 30 of them lighting up red that it's like, okay, do not click this thing so that it's able to run. And stepping back from all this a little bit, it's so annoying that so much energy is being spent holding back the forces of darkness. Look at how much we put in now to doing that. But on balance it's worth it because what can be done with computers today is truly amazing.

Leo, I think I'm going to take a sip of water, and you can tell us why we're here.

Leo: I just - maybe we should underscore, don't download drivers from third parties.

Steve: No.

Leo: Steve's a trained professional. Do not try this at home.

Steve: Good advice.

Leo: Yeah. It just makes me really squeamish.

Steve: I know. And believe me, it made me just, like, I...

Leo: Is it because they were gone? They were no longer available? Is that why?

Steve: Oh, yeah. You try to find a driver for a 1986 network interface card, I mean, it's just for DOS.

Leo: I guess you have no choice, yeah, yeah.

Steve: For DOS. For DOS.

Leo: Yeah, what are you going to do?

Steve: And it's a high level of pucker factor. And I wouldn't run them on Windows. I ran them, I moved them immediately to DOS. There's not much you could do on DOS. Basically, if a virus comes alive on DOS, it's like, oh, crap.

Leo: Yeah, I mean, yeah. Seriously. That's a virus that's been hanging out for a long time. It's pretty - that's a geriatric virus.

Steve: It's like, what's 32 bits?

Leo: I don't know what this Windows thing you're talking about is. There's a story about a mother and daughter in Nebraska who just got arrested and charged with felonies. And how did the authorities get the information? They subpoenaed their Facebook DMs.

Steve: Wow.

Leo: You know, we're really now in a situation where the government is happy to use these technologies against you. And I think we're [crosstalk] to defend ourselves.

Steve: And it's why the big question is whether encryption will be allowed to survive.

Leo: I'm sure they won't. I'm sure. They hate this. They hate this. They don't want you to be private.

Steve: It's just a matter of time. So two stories about Microsoft. One, the first is a little rough. But the second I give them some props. The rough one is last week Microsoft warned in their knowledge base article KB5017259 that "Windows devices that have the newest supported processors might be susceptible to data damage." Now, though their poorly worded title doesn't make it clear, it's not the Windows devices with the newest supported processors that might be damaged. It's users' encrypted data that has been damaged by the simultaneous use of Windows 11 with the new vector encryption instructions that are present only in the latest processors.

Microsoft's posting explained that: "Windows devices that support the newest Vector Advanced Encryption Standard" - Advanced Encryption Standard is AES, the Rijndael cipher, so this is VAES - "instruction set are susceptible to data damage. The affected Windows devices are those that use either AES XEX-based tweaked-codebook mode with ciphertext stealing" - which is AES-XTS, and that meant something to me, I'll explain it in a second - "or AES with Galois/Counter Mode." That's of course GCM, so AES-GCM. Both of those immediately raised my eyebrows since AES-GCM has become the preferred authenticating encryption mode for bulk encryption, and AES-XTS is the way data at rest is stored encrypted in mass storage.

And sure enough, in their knowledge base article, Microsoft wrote: "To prevent further data damage, we addressed this issue in May 24th, 2022 preview release and the June 14th, 2022 security release. After applying those patches," they said, "you might notice slower performance for almost one month" - and I thought, what? A month? What? - "after you install them on Windows Server 2022 and Windows 11," they said, "(original release). The scenarios that might have performance degradation include BitLocker, Transport Layer Security (TLS), and disk throughput, especially for enterprise customers."

In other words, the previously faulty encryption that was being used by Windows 11 for BitLocker and TLS communications was fast, but broken. And it was damaging its users' data. So they fixed that quickly in May and June by no longer using the VAES instructions at the cost of performance. Which is why they said you may notice things going slower, but at least your data's not being scrambled.

They wrote: "If this affects you, we strongly urge you to install May 24th, 2022 preview release or the June 14th, 2022 security release as soon as possible to prevent further damage." Then they said: "Performance will be restored after you install the June 23rd, 2022 preview release or the July 12th, 2022 security release." In other words, they're saying that they found and fixed the broken VAES implementation and have restored Windows 11 to use VAES with the most recent updates. So now you get speed; and, happily, it's not damaging your data. So this was all resolved by last month's Patch Tuesday. But there was a period, and no one really knows how long it was, where Windows 11, when used on the most modern processors which had this VAES set of instructions, the Vector AES, had broken Windows core crypto algorithms.

Microsoft wrote: "We added new code paths to the Windows 11 original release and Windows Server 2022 versions of SymCrypt to take advantage of VAES, the vectorized AES instructions. SymCrypt," they wrote, "is the core cryptographic library in Windows. These instructions act on Advanced Vector Extensions (AVX) registers for hardware with the newest supported processors."

Okay, now, obviously whatever these bugs were, they were not destroying everyone's data. Or, I mean, we would have known about that. Or perhaps it was just that very few

people were using Windows 11 original release on the very latest processors which also had these new VAES instructions. But in any event, it's a bit frightening to have this somehow escape from and to be shipped by Microsoft.

BleepingComputer carried this story, and someone commented on their story on its page by quoting the phrase "data damage" and said: "Data damage, the new marketing gloss-over for data loss and filesystem corruption." He said: "Don't be fooled. It's yet another case where Microsoft's bungled agile development practices have screwed the pooch. Their testing harnesses are entirely inadequate to support the massive legacy code bases they have to support in the time scales they need to release." And I think that's an accurate criticism, as we know. We've seen lots of evidence of this increasingly in the last couple of years. I don't think that Paul, Mary Jo, or I could have summed things up better than that.

But on the brighter side, I also have some happy Microsoft news to share. They've announced a new security offering which looks pretty good to me. It promises to provide security teams with the means to spot Internet-exposed resources in the organization's environment that they may not be aware of.

On the front page announcing this new what Microsoft calls the "Microsoft Defender" - and Paul and Mary Jo the other day said that pretty much everything was now called "Microsoft Defender." So it's Microsoft Defender this or Microsoft Defender that. They're liking that jargon a lot. Anyway, this is the "Microsoft Defender External Attack Surface Management." And on the announcement page they note the highlights.

They said: "Discover unmanaged resources: Understand the full extent of your attack surface, including shadow IT and assets created through common, everyday business growth. Multicloud visibility" - this is the first time I'd seen the term "multicloud." It's like, okay, one's not enough, let's get multiclouds. So we've got "Multicloud visibility: Maintain a dynamic inventory of external resources across multiple cloud and hybrid environments." And then, finally, "Identify exposed weaknesses: Prioritize vulnerabilities and misconfigurations hidden in unmanaged resources, then bring the resources under management to remove those exposures."

So, okay. From everything they've written, it's unclear to me, because of the terminology they've used, whether this is an external but, for example, a far more comprehensive scan, like GRC's "ShieldsUP!" service; or whether it's local network packet monitoring. If a network monitor was placed truly upstream of everything else that the enterprise was exposing, that could do the job. But it strikes me that even that could be prone to some mistakes. The focus is on mistakenly unmanaged, forgotten, or unknown network assets which might be added to the environment after, you know, they said business growth, so mergers or acquisitions, created by shadow IT, somebody's uncle plugging something in somewhere, or missing from inventory due to incomplete cataloging, or just left out due to rapid business growth. And I think all of this is a great idea.

Microsoft's Corporate VP for Security said: "The new Defender External Attack Surface Management gives security teams the ability to discover unknown and unmanaged resources that are visible and accessible from the Internet, essentially the same view an attacker has when selecting a target. Defender External Attack Surface Management helps customers discover unmanaged resources that could be potential entry points for an attacker."

And I still can't figure out exactly what it is from what they've written. As I said, the language they're using is aggravatingly imprecise. Elsewhere in describing it, they wrote: "Microsoft Defender External Attack Surface Management scans the Internet and its connections every day." Well, okay. I don't know what "the Internet and its connections" means exactly. Maybe Paul and Mary Jo will talk about this tomorrow.

Anyway, they continue. Microsoft says: "This builds a complete catalog of a customer's environment, discovering Internet-facing resources" - that sure sounds like an external scanner to me - "even the agentless and unmanaged assets. Continuous monitoring, without the need for agents or credentials" - which again sounds like outside - "prioritizes new vulnerabilities. With a complete view of the organization, customers can take recommended steps to mitigate risk by bringing these unknown resources, endpoints, and assets under secure management within the security information and event management" - and we have an acronym for that, SIEM - "and extended detection and response" - that's XDR - "tools."

So it sounds like sort of a Microsoft-offered Shodan scanner for their enterprise customers. Anyway, whatever it is and whatever it costs, in an enterprise environment where there might be too many overlapping regions of IT authority, and without any absolutely central single omniscient management, this sure seems like something that would be worth pricing out and exploring for an enterprise. It is just - it is so easy to make a mistake, and this might work to catch any such mistakes before the bad guys do. So I wanted to bring it to everyone's attention. I imagine that our enterprise-bound listeners may want to know about it.

Okay. So I'm going to talk about Dan here in a minute. First I want to do some Closing the Loop with our listeners. Someone who asked me to keep his name anonymous, so I'll just call him K.A., sent me a DM. He said: "Steve, just wanted to share my experience with 0patch as you have mentioned it a few times on Security Now!. I heartily recommend this program for anyone still having to support Windows 2008 servers or Windows 7 systems in their enterprise." Which as we know have gone out of security patch cycle unless you buy it, and it gets more expensive every year, from Microsoft. Anyway, he says, "...due to business reasons such as having to support legacy applications. Anyone can sign up for the free account to get a copy of the 0patch agent." And that's 0patch.com. And he says, "...and see in an instant what Windows modules are at risk, and how often they are being called by the OS." That's really cool.

He says: "The cost to patch a Windows system with 0patch is a fraction of what you would have to pay Microsoft, and the patches are installed instantly, with no reboot required." He said: "I was able to protect my systems within seconds of purchasing my subscription as the agent immediately implemented the patches on my machines. I hope this is helpful to our Security Now! community, as I am sure several of us would like to simply shut down non-supported Windows systems, but are unable to for business impact reasons. Please keep my handle anonymous, as I have identified myself through this message as having to support legacy systems, and I would not like anyone to trace my association with my company which could put them at risk.

"Thank you for producing my favorite podcast and equipping our community with the information we need each week to digitally protect our business and loved ones' systems." So thank you, K.A. I'm happy to share this information with our listeners. I think that the 0patch, the micropatch guys do a great job, and we're often talking about them.

Someone calling themselves Oaksong tweeted: "Does the CIA care about felonies when hiring security consultants? Do they just move them overseas?" Which struck me as a really good question. It's related, of course, to our more tongue-in-cheek Picture of the Week last week, the two paths to reaching professional security guru status. I don't remember exactly what we called it. One was the 20-year path of working your way up through the hierarchy. The other was be a hacker, get convicted, go to prison, get out in 14 months on good behavior and get hired as a security professional. So anyway, I got a kick out of that. I can certainly see both sides of it.

Simon Kirkman said: "Hi, Steve. I'm trying to set up a guest WiFi network, and I found an issue which I can't see how you got around when you did it at home." He said: "I set up a new router for visitors to our village hall, which is connected to a modem router in an office belonging to a business who are willing to allow Internet access, but not LAN access due to CCTV, et cetera, being on their network. My new router is set to a different subnet and subnet mask, and in theory is separate from the business network. But in practice, an IP address gets passed upstream to the other router, which then allows access. I can't see how to do this correctly, and it allows my guest users to access the business LAN. How did you do it with your smart home network?"

Okay, well, first of all, I did it with my smart home network by using what we affectionately, really very affectionately call "dumb routers." Mine was a smart router where the individual ports were separate network interfaces, separate NICs, not a hub or a switch, where all of those ports were all on the same LAN. That allowed me to assign different LANs to different ports. But there is the famous "three dumb routers" solution that I developed years ago on this podcast. It's a way of creating two mutually-isolated networks using three simple and standard NAT routers and pretty much zero configuration. The three routers are wired in a "Y" configuration. We'll call it Router A connects to the Internet on its WAN interface and provides Internet service to Routers B and C by connecting each of their WAN interfaces to two of Router A's LAN interfaces. Thus the "Y" connection.

A useful simplification for simple NAT routers is to think of them as a one-way valve where traffic can easily leave the network, passing out of the router toward the Internet, but unsolicited and unexpected traffic cannot enter the network. In fact, we now utterly depend upon this feature of NAT routers to act as our Internet firewalls for us. This same principle works for the "Y" configuration to prevent any traffic from one of the LANs from having any access whatsoever into the other LAN because each LAN is protected by its own dumb router which is acting as a one-way valve.

Now, that requires three routers, each of which can be dumb. So, speaking to Simon, if it's feasible for you to place a third router upstream on the WAN side of both the business router and your village hall router, that would provide perfect isolation. But there is a two-router variation which might also work. Switch the roles of the two routers. Place the village hall router on the Internet and connect the business router which requires the privacy to one of the LAN ports of the village hall router. In that way, the traffic on the business's LAN is protected by its router's one-way NAT firewall, and the village hall router that doesn't need that protection doesn't need that. It might still be able to see the village hall traffic, but not the other way around. So in other words, three dumb routers are needed for two-way privacy; but two routers can be used when one-way privacy is sufficient.

So as it happens I sent this answer back to Simon, who later replied: "Hi, Steve. Thanks for that. I have a third router around. I'll look to set that up. Had not thought about doing it that way. Thanks very much."

Leo: It's nice that you can do it all wireless. That's cool.

Steve: Yes, yes. Jose C. Gomez. He said: "Hi, Steve. Here's a pretty complete demo and explanation of how Passkeys are going to work and interface between Microsoft, Google, and Apple, presented by some of their product managers and engineers." And in his tweet he sent me a YouTube link. I turned it into this week's shortcut. He finished his tweet saying: "Looks like there is no Passkey sharing at all, and it's more cross-device auth and recreate."

So for those who are interested in seeing this working, the 14-minute YouTube video posted by the FIDO Alliance on their YouTube channel is very good. It has a blessedly brief introduction by a FIDO marketing person, followed by brisk walk-throughs by Google and Microsoft product managers. So as I said, the video is this week's shortcut of the week, meaning that it's grc.sc/883. That'll bounce you to the YouTube link.

So the short version is that it's all exactly what we thought. And they do succeed in making it all look wonderful. The Google guy highlights the "magic" created by Apple's iCloud synchronization, such that a Passkey created on one Apple device will be known to all of your other Apple devices. And they show how a Bluetooth-enabled phone and a Bluetooth-enabled desktop can use a QR code displayed on a web page to allow the phone to authenticate the user on that desktop; and how the user, now having been authenticated, may then choose to create another Passkey locally on that device, that is, that desktop device, so that future logins can be done natively without the phone.

As we know, it's not as good as we could have had. And bridging isolated brands such as Apple, Google, and Microsoft will - this pretty much confirms it - require creating multiple functionally duplicate Passkeys for every website on every machine that lacks a means of synchronizing and sharing existing Passkeys. But it's the system we're going to get. And thanks mostly to the authentication automation which WebAuthn brings to finally create an alternative to the kludge of clunky form-fill-in authentication, it will be better than what we've had.

But implementing it on the server side, as we know, still requires some major work individually from each and every website. That's where the form-fill password managers excelled is that the websites were spoiled. They didn't have to change anything. Everything was done by filling in the form. But it's a mess. So it's going to be very interesting to see how all this transpires over the next few years. But our initial impression is confirmed in this 14-minute video. But it also, I have to say, I mean, they make it look breezy and not burdensome. It's just not as nice as it could have been.

Leo: Yeah.

Steve: @ElectronicAthro said: "Hi, Steve. Love Security Now! and recently came across something interesting I thought I'd ask you about. At least I thought it was interesting. Maybe it's trivial, and I should know better." He said: "I recently took a United flight, and they allow their in-flight WiFi to be used for IP-based messaging apps for free. However, they block the sending of images. So how do they detect that one is sending an image if one is using a messaging app that has strong encryption like Signal? I would have thought that since Signal does the encryption at the 'end,' any image sent by a Signal message would be indistinguishable from a text message. But after attempting to send an image, United has clearly figured out how to detect an image in a message and block it. How is this possible while maintaining Signal's encryption?"

Okay. My guess would be that it's all about bandwidth usage and size. Text is truly tiny, whereas any image is huge, massive by comparison. So it would be easy to simply watch each user on the airplane for the rate of data that they are exchanging; and, if it exceeds some very low maximum, which is all texting is ever going to be, right, like writing some text and hitting SEND is a little tiny little blurch of bandwidth use, it would be very easy for them to set a very low maximum and, if they exceed that, cut them off. And also note that a simple bandwidth cap, a very low bandwidth cap is also what United or any carrier would want. It's sort of a nice compromise. Their travelers can trickle out, in and out, text for free, so long as it's at such a low bandwidth as to be insignificant to them. But if you want the cap lifted, fork over some cash.

Relief Twitcher tweeted: "@SGgrc You're a lifesaver! Today, a PC in a pharmacy that I support installed Microsoft Update KB5014666, the one that makes the duplicate USB printer and deselects the port for the original. I'm not sure why my organization let the patch through, but this PC shares a vital label printer with the rest of the pharmacy, and suddenly no one could print labels. As soon as I saw the duplicate printer with the '(Copy 1)' in its name, I knew exactly what do. The reason I knew was because I had listened to Security Now! 881. You saved me a lot of work and my pharmacists a lot of down time. Thank you."

Oh, and he said: "While I have your attention, I'll make another plea for you to read 'Sea of Tranquility' by Emily St. John Mandel." He says: "I found it to be compelling speculative fiction. I think they call it that because it's light on the science-y details, opting instead to just concede that things like time travel and domed cities exist, and to focus instead on the story of the humans in that environment." He says: "I think you would like it."

So first, it's very cool when something from the podcast so nicely lines up with a real-world event. And as for his book recommendation, since I'm currently without an engaging science fiction novel, I purchased the book for $11 for my Kindle. And I'd have to say it has endless amazing over-the-top reviews there. So it looks like an interesting possibility. My nephew, who is similarly hooked on Ryk Brown's Frontiers Saga, and who with me as been waiting for the next one to drop, has not yet discovered Peter Hamilton. So he's heading into "Fallen Dragon," which I'm sure he's going to go nuts over, and then I'm going to aim him at "Pandora's Star." So I envy him for not yet having found Hamilton because, boy, does he have some great stuff ahead.

Martin Rojas. He said: "My sister had a severe allergic reaction" - he said she is fine - "but in talking to the paramedics we talked about medical bracelets or ID cards. She has a complex medical history and medication, as do many of the people they pick up. The paramedic mentioned that if there was a card with NFC or a QR code they could scan that information, and it would be a great help to them in an emergency." He says: "That part is easy, but it would also be public for anyone to scrape the info. My question is whether there is some pattern that could serve both as secure, but also easy to access by emergency personnel. I was thinking password printed in the card, but maybe there is something better."

From a theoretical privacy protection standpoint, I had two thoughts in response to Martin. The first would be to have a publicly accessible QR code that anyone could use to access medical records, which would be carried by the person, but in such a way that it was not readily accessible to anyone. For example, make it a comfortable silicone wristband that's never removed, which identifies itself as offering critical emergency information on its underside only. Thus it provides a degree of physical privacy by physically limiting the circumstances under which someone could obtain access to the QR code. It could also make very clear what the person's most important health requirements are, which would be spelled out in the region's most common language on the underside so that you wouldn't have to even scan the QR code. You could immediately determine what was most important, and then the QR code could provide additional backup.

The second solution, which could be applied as an additional layer of privacy protection if required, would be for the QR code, which would presumably take anyone who scans it to an emergency information supply service, that could also require login authentication by an accredited and confirmed emergency services supplier. While that would offer greater privacy protection, the worry would be that the information not be made available as quickly as it could be or maybe at all if the authentication failed, or if the provider of the service didn't have an account with the information provider.

So I think if it was me, I'd worry less about privacy and more about being certain that any special medical needs, allergies, et cetera, were readily known to someone who needed to obtain them.

Leo: And let me put in a plug here for turning on Medical ID. If you have an iPhone, and all first responders know this, go to the Health app, scroll down, set up Medical ID. Mine is set up with my medications, my allergies, everything that first responders would need to know. The most important thing a first responder needs, I'm told by first responders, is a number to call for your next of kin or, well, that sounds like you're dead. Your nearest, you know, your contact, emergency contact, because they often will just use that. But there is that information. iPhones have it built in. They know how to get to it, if you have to press and hold the side button and the volume up button at the same time and swipe the Medical ID slider. So it's not something a casual privacy thief would get to.

Android unfortunately does not have this built in on all versions of Android, but there are third-party apps. JointCommission.org has the information. There's one called Medical ID on the Google Play Store. Use those because first responders know, who doesn't have a smart phone? They know immediately to go for the smart phone. So I think Medic Alert bracelets and necklaces and other things like that have been superseded these days by your smart phone. So if you have a smart phone, turn that on. It's really important.

Steve: Very cool. And I guess I would argue, even if you don't have any medical issues, but you do want an emergency responder to be able to get a hold of the person that you need to have notified.

Leo: Yes. Most important information of all. Exactly.

Steve: Yeah. Okay. So finally, Dan Bernstein sues the NSA. As I mentioned at the top, Dan is one of my favorite cryptographer mathematicians since he's the father of the most efficient 25519 family of elliptic curve crypto and a number of other core crypto primitives that I adopted for SQRL and which have subsequently been adopted for use by TLS and even optionally by WebAuthn that would make it available for Passkeys. And coincidentally, Dan and I independently came up with the idea that was - it's called SynCookies, as a way to prevent resource depletion in TCP/IP stacks which are caused by SYN flooding attacks. The idea was a way to encode the important details of the SYN packet in the replying SYN/ACK so that stateless connection setup became possible.

Anyway, Dan was born in 1971, so he was 24 years old when, as a student at UC Berkeley, he brought his first lawsuit against the United States. Dan's 50 years old now. He wanted to publish a paper at the time with its associated source code on his - Snuffle was the name of it, the Snuffle encryption system. But that would have been illegal at the time, so he sued and won. He sued the United States and won. After four years and one regulatory change, the Ninth Circuit Court of Appeals ruled that software source code was speech protected by the First Amendment, and that the government's regulations preventing its publication were unconstitutional. And we owe Daniel for that.

I'm bringing this up today, and I called that "Dan's first lawsuit" because as I mentioned already, last Friday he announced that he has now sued the NSA in a blog posting titled: "NSA, NIST, and post-quantum cryptography: Announcing my second lawsuit against the U.S. government."

Leo: It worked once before.

Steve: His blog post is lengthy, and I want to read, digest, and research the entire thing. So unless something more interesting pops up before next week, it will likely be next week's topic. For anyone who doesn't want to wait for me, the link to Dan's blog post is in the show notes. So I think that's what we'll be talking about next week. And Leo, after our final sponsor break, we're going to talk about somebody I just discovered who you have known about for quite a while, and a really compelling piece of his writing.

Leo: Oh, good. Can't wait. And I put into the chatroom and the show notes, well, no, yours are the show notes, but I put into the chatroom and the Discord information about turning on Medical ID. Many Android phones, unless my Samsung Galaxy phone has that built-in, otherwise they're third-party apps. And all iPhones to my knowledge support that.

Steve: Here's what happened. When I settled down late yesterday morning to begin assembling today's podcast, I started by catching up with my week's past Twitter DMs. The first and most recent DM I encountered was from a listener named Theron Keller who pointed to something that astounded me, and I thought it was so important that it became today's topic. He tweeted: "Hi, Steve. I'm a few weeks behind on SN. I just heard the episode where you mentioned coding all night long. Then today I saw this and of course thought of you. I'm sure other coders would agree."

So he pointed to a posting on Facebook where someone had apparently just discovered something someone else had written back in July of 2009. After scanning the Facebook posting I followed the source reference link to the original content, and thus stumbled upon the work and writings of someone I had never been very much aware of. The guy's name is Paul Graham. And here's a very brief bio of Paul that could clearly be much longer.

He is a programmer, writer, and investor. In 1995, he and Robert Morris - yes, that Robert Morris - started Viaweb, the first software-as-a-service company. That's 1995. And I believe it was a Lisp-based storefront-creating service. So Viaweb was acquired by Yahoo in 1998, where it became Yahoo Store. In 2001 Paul started publishing essays at PaulGraham.com, which now gets around 25 million page views per year. In 2005 he and Jessica Livingston - now his wife - Robert Morris, and Trevor Blackwell started Y Combinator, the first of a new type of startup incubator. Since 2005 Y Combinator has funded over 3,000 startups, including Airbnb, Dropbox, Stripe, and Reddit. In 2019 he published a new Lisp dialect written in itself called Bel. Paul is the author of "On Lisp" published by Prentice Hall in '93.

Leo: Own it.

Steve: "ANSI Common Lisp," Prentice Hall '95.

Leo: Own it.

Steve: And "Hackers & Painters," of all things, which was published by O'Reilly in 2004.

Leo: I own all three, yeah.

Steve: Uh-huh. He has his Bachelor's degree in Philosophy from Cornell; his Master's and a Ph.D. in Computer Science from Harvard. He's also studied painting at the Rhode Island School of Design and at the Accademia di Belle Arti in Florence. The well-known technology journalist Steven Levy has described Paul as a "hacker philosopher"; and, given what I've seen, I would tend to agree. I was curious about his Ph.D. dissertation, so I tracked it down.

Leo: Wow. I have not read that.

Steve: It is quite something, Leo. It's titled - I love the title - "The State of a Program and Its Uses." It's wonderfully mystical. I read the abstract. As you might expect, it's some seriously nice pure computer science thinking.

So poking around a bit more, I was getting intrigued by this guy. I looked at a couple of his Twitter postings. A recent post of his from Saturday, three days ago, he tweeted: "In office hours today" - and I should mention, as we'll see, that the use of this term "office hours" is important to him. He says: "I talked to a pair of founders who needed a new idea. It turned out they already had a great idea, but had been ignoring it because they didn't know how to monetize it. I told them to just build it. This thing could have 100 million users."

And yesterday he tweeted - I love this. "Effective organizations are unnatural. The natural state of organizations is bureaucracy and turf wars, and once deprived of effective leadership they revert to their natural state with shocking speed."

Leo: Oh, boy, is that true. Holy cow.

Steve: Isn't that great?

Leo: Yes.

Steve: Isn't that great? Oh. And looking a bit further back, on August 1st, he tweeted: "The hardest people for founders to hire are so-called C-level executives because these people are the best fakers in the world."

Leo: Aha, yes.

Steve: He said: "Even the best founders make absolutely disastrous mistakes hiring these people. It happens far more often than anyone realizes."

Leo: Oh, yeah, it really does.

Steve: "Because neither party wants to talk about it. So after nearly destroying one company, the exec cheerily goes off to their next opportunity." And of course actually

this put me in mind of someone I've talked about on the podcast before, a horrible person by the name of Ron Posner, who Peter Norton hired because Peter thought he needed like an executive.

Leo: Big mistake.

Steve: To run stuff. Oh, boy, yeah. So anyway, if you're into following people on Twitter, Paul might be someone worth following. I don't follow anyone on Twitter, but I'm really tempted to follow him. He tweets as @paulg, and you'd be joining his 1.5 million current followers. And I'm unsure why I find this guy so fascinating. That doesn't happen that often. He's got something.

So it seems pretty clear that in Paul Graham we have a serious computer science guy with a strong creative side and a very strong entrepreneurial business side. And that might be what's hooking me. He made money early in the run-up of the Internet and the dot-com revolution. It also appears that he's one of those still rarer guys who didn't make it by chance, by being in the right place at the right time, but then never able to again recreate that first early success. He's a serially successful entrepreneur. And he's either spent a lot of time thinking, or he's very good at it. And it turns out that Paul is also an outstanding writer, which brings us to today's topic.

As I said earlier, as I began reading what Paul wrote, its subject and content resonated so deeply with me as I know it will with so many of this podcast's listeners that I knew that sharing it here would be the best possible use of everyone's time this week. It helped that there was not a huge amount of compelling security industry news this week. But I had already made the decision to share this as this week's topic before I even knew that. He gives explicit permission for his essays to be included, in full, in school newspapers and the like, asking that the URL to its original page be included, as I've already done several times in these notes.

So here's what Paul Graham wrote just over 13 years ago, in July of 2009, under the title "Maker's Schedule, Manager's Schedule." He said: "One reason programmers dislike meetings so much is that they're on a different type of schedule from other people. Meetings cost them more. There are two types of schedule, which I'll call the manager's schedule and the maker's schedule. The manager's schedule is for bosses. It's embodied in the traditional appointment book, with each day cut into one-hour intervals. You can block off several hours for a single task if you need to, but by default you change what you're doing every hour. When you use time that way, it's merely a practical problem to meet with someone. Find an open slot in your schedule, book them, and you're done.

"Most powerful people are on the manager's schedule. It's the schedule of command. But there's another way of using time that's common among people who make things, like programmers and writers. They generally prefer to use time in units of half a day, at least. You can't write or program well in units of an hour. That's barely enough time to get started. When you're operating on the maker's schedule, meetings are a disaster. A single meeting can blow a whole afternoon by breaking it into two pieces each too small to do anything hard in. Plus you have to remember to go to the meeting. That's no problem for someone on the manager's schedule. There's always something coming on the next hour; the only question is what. But when someone on the maker's schedule has a meeting, they have to think about it.

"For someone on the maker's schedule, having a meeting is like throwing an exception. It doesn't merely cause you to switch from one task to another; it changes the mode in which you work. I find," he writes, "one meeting can sometimes affect a whole day. A meeting commonly blows at least half a day by breaking up a morning or an afternoon.

But in addition there's sometimes a cascading effect. If I know the afternoon is going to be broken up, I'm slightly less likely to start something ambitious in the morning. I know this may sound oversensitive, but if you're a maker," he says, "if you're a maker, think of your own case. Don't your spirits rise at the thought of having an entire day free to work, with no appointments at all?" Oh, my lord, yes. "Well, that means your spirits are correspondingly depressed when you don't. And ambitious projects are by definition close to the limits of your capacity. A small decrease in morale is enough to kill them off.

"Each type of schedule works fine by itself. Problems arise when they meet. Since most powerful people operate on the manager's schedule, they're in a position to make everyone resonate at their frequency if they want to. But the smarter ones restrain themselves, if they know that some of the people working for them need long chunks of time to work in it.

"Our case is an unusual one." Speaking of Y Combinator. He says: "Nearly all investors, including all venture capitalists I know, operate on the manager's schedule. But Y Combinator runs on the maker's schedule. Rtm" - that's Robert Morris - "and Trevor and I do because we always have; and Jessica does too, mostly, because she's gotten into sync with us.

"I wouldn't be surprised if there start to be more companies like us. I suspect founders may increasingly be able to resist, or at least postpone, turning into managers, just as a few decades ago they started to be able to resist switching from jeans to suits. How do we manage to advise so many startups on the maker's schedule? By using the classic device for simulating the manager's schedule within the maker's: office hours. Several times a week I set aside a chunk of time to meet founders we've funded. These chunks of time are at the end of my working day, and I wrote a signup program that ensures all the appointments within a given set of office hours are clustered at the end. Because they come at the end of my day, these meetings are never an interruption. Unless their working day ends at the same time as mine, the meeting presumably interrupts theirs. But since they made the appointment, it must be worth it to them. During busy periods, office hours sometimes get long enough that they compress the day, but they never interrupt it.

"When we were working on our own startup, back in the '90s, I evolved another trick for partitioning the day. I used to program from dinner until about 3:00 a.m. every day because at night no one could interrupt me. Then I'd sleep till about 11:00 a.m. and come in and work until dinner on what I called "business stuff." I never thought of it in these terms, but in effect I had two workdays each day, one on the manager's schedule, and one on the maker's.

"When you're operating on the manager's schedule you can do something you'd never want to do on the maker's. You can have speculative meetings. You can meet someone just to get to know one another. If you have an empty slot in your schedule, why not? Maybe it will turn out you can help one another in some way. Business people in Silicon Valley, and the whole world for that matter, have speculative meetings all the time. They're effectively free if you're on the manager's schedule. They're so common that there's distinctive language for proposing them, saying that you want to 'grab coffee,' for example.

"Speculative meetings are terribly costly if you're on the maker's schedule, though. Which puts us in something of a bind. Everyone assumes that, like other investors, we run on the manager's schedule. So they introduce us to someone they think we ought to meet, or send us an email proposing we grab coffee. At this point we have two options. Neither of them are good. We can meet with them and lose half a day's work, or we can try to avoid meeting them and probably offend them.

"Till recently we weren't clear in our own minds about the source of the problem. We just took it for granted that we had to either blow our schedules or offend people. But now that I've realized what's going on, perhaps there's a third option, to write something explaining the two types of schedule. Maybe eventually, if the conflict between the manager's schedule and the maker's schedule starts to be more widely understood, it will become less of a problem. Those of us on the maker's schedule are willing to compromise. We know we have to have some number of meetings. All we ask from those on the manager's schedule is they understand the cost."

So I just thought that the crystallization of that is what hit me so clearly. And I heard you getting it at the beginning of this, Leo.

Leo: Oh, yeah. Oh, yeah.

Steve: What I like so much about what Paul has created here is the clarity of the distinctions he's made. I've often explained to my friends and family about what I call the - I call it "switching cost." It is so much more efficient to stay on one thing until it's finished than to switch back and forth among multiple things. In programming we call it "context switching," and it's expensive there, too.

And I definitely operate on a Maker's schedule, and I always have. A day with no scheduled interruptions planned is joyous. It's a block of unbroken time I savor in anticipation and actively appreciate. And it completely explains why I was so miserable running a company with 23 employees, which was GRC's size at its peak. I liked every single employee I had. I enjoyed them as individuals. But I was worried that they were going to need something from me when I was in the middle of doing something else. And the point is I am always in the middle of doing something else. You know? And I want to be. That's the way I want to spend my life. Pretty much most of the time I just want to be left alone to work. And Leo, I sort of feel I have a kindred spirit in you.

Leo: Yeah.

Steve: If I'd had the wisdom back when GRC was a cauldron of chaos, I should have worked at home in seclusion four days a week and held office hours, as Paul does, on Fridays. On those Fridays I would have pre-resigned myself ahead of time that I would not be getting any code written that day, and perhaps I would have both been more available and less miserable and grumpy.

So anyway, I can imagine many of the Maker's Schedule listeners we have here maybe sending a link to this little piece of writing to their managers, asking and pleading with them to read it. If I were drowning in a corporate setting, I think I would do so.

Leo: I am going to recommend another one which I blogged about a couple years ago.

Steve: Oh, good.

Leo: It goes along with this one, Steve, that he wrote in 2016, called "Life Is Short." Because it's the same premise extended even farther, you know. Don't waste time on B.S. because life is short.

Steve: Yeah. So Paul's articles page is at PaulGraham.com/articles.html. It lists this and 211 other short essays. I've only poked around at them a bit, but I find something about them to be quite compelling.

Leo: Yeah. Once you start reading, you won't stop.

Steve: Yes. I did not want to stop reading them, but I had this podcast to finish. And I think before I start in on this next book I'm going to spend some time there. At the top of his page he says: "If you're not sure which to read, try 'How to Think for Yourself,' or 'Do Things That Don't Scale,' or 'How to Lose Time and Money.'" And of course we have Leo's recommendation, "Life Is Short."

Leo: "Life Is Short." He also wrote THE article on why you should learn Lisp. So I can't help but recommend that, as well.

Steve: Oh, cool. I will be getting to it.

Leo: You will.

Steve: I leave our listeners in Paul Graham's very capable hands.

Leo: Yeah. He's amazing. A great thinker and a great writer. And those two, when you get the two together, fantastic. Fantastic.

Steve: Yeah.

Leo: All right. I think that means you're going to go back to work, and I'm going to go back to reading, and the rest of you, well, you could keep listening to podcasts. That's my suggestion. Steve Gibson does this show every Tuesday, 11:00 a.m. Pacific, I'm sorry, 1:30 p.m. Pacific, that's 4:30 p.m. Eastern, 20:30 UTC. If you want to watch us do it live, that's why I say the time, so you can watch it live at TWiT.tv/live. There's audio and video there. Chat with us live at irc.twit.tv or in the Club TWiT Discord.

On-demand versions of the show available at our site, TWiT.tv/sn. We've got 64Kb audio and video available there. And of course you can subscribe to it in your favorite podcast client, as well. Steve has two unique versions of it at his website, GRC.com. He's got a 16Kb version for the bandwidth-impaired. He also has really good transcripts written by a human, Elaine Farris, so you can read those and follow along, or use them to search for parts. All of that is at GRC.com.

While you're there, pick up SpinRite, Steve's bread and butter. We've got to keep him in bread and butter. All you have to do right now, get 6.0. You will participate in

the development of 6.1 and get it for free if you buy today, the minute it's available. GRC.com for SpinRite, the world's best mass storage maintenance and recovery utility. Lots of other free stuff there. You can leave a message for Steve there, GRC.com/feedback. But probably more useful to leave a DM for him. His DMs are open on Twitter. He's @SGgrc. You can DM him there, @SGgrc. I think that covers everything, Mr. G.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/