Security Now! #883 - 08-09-22

The Maker's Schedule

This week on Security Now!

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 user's 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.

Security News

Crypto is Hard

In fact, it's even harder than we know. 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 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 (CSIS) at KU Leuven managed to break one of those four late-stage candidate algorithms for post-quantum encryption. Fortunately, the dilithium crystals are still intact. The algorithm they cracked, SIKE — the abbreviation for Supersingular Isogeny Key Encapsulation — made it through all of the earlier stages of the US Department of Commerce's National Institute of Standards and Technology (NIST) competition.

NIST's Post-Quantum Crypto replacement campaign has been running for five years.

- 1st round (2017) — 69 candidates
- 2nd round (2019) — 26 surviving candidates
- 3rd round (2020) — 7 finalists, 8 alternates
- 4th round (2022) — 3 finalists and 1 alternate were selected to become the standards.

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 from a pure math angle, attacking the core of the algorithm's design instead of any potential code vulnerabilities.

SIKE's base encryption was based upon something known as Supersingular Isogeny Diffie-Hellman (SIDH). But the mathematicians were able to show that SIDH was, in turn, vulnerable to a mathematical attack which was developed in 1997 known as the "glue-and-split" theorem which is an attack on elliptic curves. In response to 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."* And he added that all of this goes back to cryptographers' sometimes imperfect dominion of pure mathematics, in that the approach taken by the researchers was *"really unexpected."* Uh huh.

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.

And they received a bounty award of $50,000 from Microsoft for their research which they published in a paper titled: "An Efficient Key Recovery Attack on SIDH."

The people who have been watching and participating in this process are concerned that this appears to be as difficult as it is. In a note to Ars Technica which they 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 3rd round of the NIST review process, before being completely broken using a classical algorithm." He added that: *"Three of the four PQC 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/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. We've already been doing this for 5 years. Thank goodness that as an industry we started as early as we did, and that we are now focusing upon this as intently as we are. Even though sufficiently powerful quantum computers still seem to be a safe distance away (have they factored the number 31 yet?) it's increasingly 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 quantum computing emerges. At some point, it might even be worth upgrading existing stored encrypted data at rest from pre-quantum encryption to post-quantum encryption. Read it, decrypt it under pre-quantum, re-encrypt it under post-quantum, and store it back.

VirusTotal: Deception at a scale

Last week VirusTotal published a report titled "Deception at Scale" where they laid out the terrain of the malware samples that are uploaded to them for their analysis. They sit in the perfect place to see what's going on.

I've complained in the past that signing my executables wasn't sufficient proof of the integrity of my apps to bypass various hair-triggered malware cautions. But VirusTotal reported, among other things, 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. So, 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 publisher. Moving forward, the only thing that can be used is the reputation that the hash of a given executable earns over time. Any new hash will need to start over from scratch earning the reputation that that **exact** code file is trustworthy.

If you care to protect yourself by inspecting the Certificate Authority who issued the authenticode certificate that was used to sign a program you're considering running, it's worth noting that 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 immediately familiar, it's probably because they renamed themselves after their repeated conduct soiled and spoiled their previous name, which was Comodo. I imagine that they're the favorite of malware authors 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 CA. It's roughly analogous to what Let's Encrypt did to TLS connections. Once upon a time, having a web server certificate meant something. Not anymore. Today, everyone needs to have one, and they mean nothing.

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 as a means to abuse the well-earned trust they've earned with users everywhere. While those are the top three, the top

10 are rounded out by 7-Zip, TeamViewer, CCleaner, Edge, Steam, Zoom, and WhatsApp.

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."*

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 to get around IP-based firewall defenses. Some of the most abused domains 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.

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."

Ultimately, checking anything that's downloaded which might be suspicious against VirusTotal is, I think, the best thing one can do. As I mentioned a while ago, when I was needing to bring old DOS machines onto my network 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.

It's so annoying that all of this energy is being spent holding back the forces of darkness. But on balance it's worth it. What can be done with computers today really is amazing.

Windows 11 might damage encrypted data

Speaking of pre-Quantum encryption, 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"*.

https://support.microsoft.com/en-gb/topic/kb5017259-windows-devices-that-have-the-newest-supported-processors-might-be-susceptible-to-data-damage-d5e7c0cb-6e0a-4865-81ed-c82e91657a24

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 user's encrypted data that HAS BEEN DAMAGED by the simultaneous use of Windows 11 and the new vector encryption instructions that are present in the latest processors.

Microsoft's posting explained that: *Windows devices that support the newest Vector Advanced Encryption Standard (AES) (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 (AES-XTS)" or "AES with Galois/Counter Mode (GCM) (AES-GCM)"

Those both 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 the May 24, 2022 preview release and the June 14, 2022 security release. After applying those updates, you might notice slower performance for almost one month after you install them on Windows Server 2022 and Windows 11 (original release). The scenarios that might have performance degradation include:*
>
> - *BitLocker*
> - *Transport Layer Security (TLS) (specifically load balancers)*
> - *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 user's data. So we fixed that quickly in May and June by no longer using the VAES instructions at the cost of performance. They wrote...

> *If this affects you, we strongly urge you to install the May 24, 2022 preview release or the June 14, 2022 security release as soon as possible* ***to prevent further damage****. Performance will be restored after you install the June 23, 2022 preview release or the July 12, 2022 security release.*

In other words, we found and fixed the broken VAES implementation and have restored Windows 11's use of VAES with the most recent updates. So, this was all resolved by last month's Patch Tuesday, but there was a period where Windows 11 on the most modern processors had broken its core crypto algorithms. Microsoft said:

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

Now, obviously whatever these bugs were, they were not destroying everyone's data. Or perhaps it was just that few people were using Windows 11 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 by writing:

"Data damage" the new marketing gloss over for "data loss" and "filesystem corruption". 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.

Neither Paul, Mary Jo nor I could probably have summed things up better than that.

Microsoft Defender External Attack Surface Management

I also have some happy Microsoft news. Microsoft has announced a new security offering which promises to provide security teams with the means to spot Internet-exposed resources in their organization's environment that they might not be aware of.

https://www.microsoft.com/en-us/security/business/cloud-security/microsoft-defender-external-attack-surface-management

On the front page announcing the new *"Microsoft Defender External Attack Surface Management"* they note the highlights:

Discover unmanaged resources: *Understand the full extent of your attack surface, including shadow IT and assets created through common, everyday business growth.*

Multicloud visibility: *Maintain a dynamic inventory of external resources across multiple cloud and hybrid environments.*

Identify exposed weaknesses: *Prioritize vulnerabilities and misconfigurations hidden in unmanaged resources, then bring the resources under management to remove those exposures.*

It's unclear to me whether this is an external, but 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 could do the job. But the focus is on mistakenly unmanaged, forgotten or unknown network assets which might be added to the environment after mergers or acquisitions, created by shadow IT, missing from inventory due to incomplete cataloging, or left out due to rapid business growth. **And... it's 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."*

I still can't figure out exactly what it is from what they've written because 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. This builds a complete catalog of a customer's environment, discovering internet-facing resources—even the agentless and unmanaged assets. Continuous monitoring, without the need for agents or credentials, 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 their security information and event management (SIEM) and extended detection and response (XDR) tools.

It does sort of sound like 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 omniscient management, this sure seems like something that would be worth pricing out and exploring.

It's just SO EASY to make a mistake and this might work to catch any such mistakes before the bad guys do.

Closing The Loop

"K.A." sent by DM:

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. (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 see in an instant, what Windows modules are at risk and how often they are being called by the OS. 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! 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 not able 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 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!

oaksong @oaksong

Does the CIA care about felonies when hiring security consultants? Do they just move them overseas?

That's a really good question. It related, of course, to our more tongue-in-cheek picture of the week. I wonder what the CIA's actual hiring practices would be in that regard. I can certainly see both sides of it, which is what makes it such an interesting question.

Simon Kirkman @simonkirkman

Hi Steve, I'm trying to set up a guest wi-fi network, and I've found an issue, which I can't see how you got around it when you did it at home...

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 etc being on their network. My new router is set on a different subnet and subnet mask and in theory is separate to 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?

This brings us back to 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 near zero configuration. The three routers are wired in a "Y" configuration. Router "A" connects to the Internet on its WAN interface and provides Internet service to router's "B" and "C" by connecting each of their WAN interfaces to two of router "A"s LAN interfaces.

A useful simplification for simple NAT routers is a one-way valve where traffic can easily leave the network, 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. This same principle works for the "Y" configuration to prevent any traffic from one of the LANs from having any access whatsoever to the other LAN.

That requires three routers, each of which can be dumb. So, 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 that 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 valve NAT firewall. It might 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. :)

I sent this explanation back to Simon, who later replied:

Hi Steve, thank you for that, I have a third router around, I'll look to set that up, hadn't thought about doing it that way! Thank you very much Simon

Jose C Gomez / @joc85

Hi Steve here is 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. https://www.youtube.com/watch?v=SWocv4BhCNg

looks like there is no pass key Sharing at all and it's more cross device auth and re create 😑

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. The video is this week's shortcut of the week: https://grc.sc/883

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 devices. And they show how a bluetooth-enabled phone and desktop can use a QR code displayed on a web page to allow the phone to authenticate the user. And how the user, now authenticated, may then choose to create another Passkey locally on **that** 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** 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 to the authentication automation which WebAuthn brings, to finally create an alternative to the kludge of clunky form-fill authentication, it will be better than what we've had. But implementing it on the server side still requires some major work. So it's going to be really interesting to see how that transpires.

@ElectronicAthro

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!) I recently took a United flight, and they allow their inflight 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?

My guess would be that it's 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 for the RATE of data they are exchanging and if it exceeds some very low maximum, cut them off. And also note that a simple bandwidth clamp is exactly what they want. It's sort of a nice compromise: Their travelers can trickle out text for free, so long as it's at such a low bandwidth as to be insignificant. But if you want the cap lifted, fork over some cash.

ReliefTwitcher / @ReliefTwitcher

@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 SN881. You saved me a lot of work and my pharmacists a lot of down time--thank you!

(While I have your attention, I'll make another plea for you to read "Sea of Tranquility" by Emily St. John Mandel. I found it to be compelling "speculative fiction." I think they call it that because it's light on the "sciency" 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. I think you would like it.

It's very cool when something from the podcast so nicely matches up with a real-world event!

And as for 'ReliefTwitcher's book recommendation, since I'm currently without an engaging science fiction novel, I purchased the book for $11 for my Kindle. The endless amazing over-the-top reviews for it are stunning, so this certainly looks like an interesting possibility. My nephew, who is similarly hooked on Ryk Brown's Frontiers Saga, hasn't yet discovered Peter Hamilton... so he's heading into Fallen Dragon before reading Pandora's Star. I envy him not yet having found Hamilton.

martin rojas / @martinrojas

My sister had a severe allergic reaction (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 QR code they could scan with that information it would be a great help in an emergency. 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 card, but maybe there is something better?

From a theoretical privacy protection standpoint I have two thoughts:

The first would be to have a publicly-accessible QR code, that anyone could use to access medical records, be carried by the person, but in such a way that it is 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. I could also make very clear what the person's MOST important health requirements are spelled out in the region's most common language.

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, 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 might not be made available as quickly as it could be or at all if the authentication failed or if the provider of the service didn't have an account with the information provider.

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

Daniel Bernstein sues the NSA
http://blog.cr.yp.to/20220805-nsa.html

Dan Bernstein is one of my favorite cryptographer/mathematicians since he's the father of the most efficient 22519 family of elliptic curve crypto and a number of other core crypto primitives that I adopted for SQRL and which has subsequently been adopted for use by TLS and optionally WebAuthn. Dan and I also independently came up with the idea of "SynCookies" as a way to prevent resource depletion in TCP/IP stacks 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 was possible.

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 wanted to publish a paper with its associated source code on his Snuffle encryption system. But that would have been illegal at the time, so he sued 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.

I'm bringing this up today, and I called that "Dan's first lawsuit" because 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."*

Dan's 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.

The Maker's Schedule

http://www.paulgraham.com/makersschedule.html

When I settled down late yesterday morning to begin assembling today's podcast, I started by catching up with my past week's Twitter DM's. The first, and most recent DM I encountered was from a listener Theron Keller who pointed to something that astounded me, and I thought it was so important that it became today's topic. Theron 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!*

Theron 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:

> Paul Graham is a programmer, writer, and investor. In 1995, he and Robert Morris started Viaweb, the first software as a service company. Viaweb was acquired by Yahoo in 1998, where it became Yahoo Store. In 2001 he started publishing essays on https://paulgraham.com, which now gets around 25 million page views per year. In 2005 he and Jessica Livingston, Robert Morris, and Trevor Blackwell started Y Combinator, the first of a new type of startup incubator. Since 2005 Y Combinator has funded over 3000 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** (Prentice Hall, 1993), **ANSI Common Lisp** (Prentice Hall, 1995), and **Hackers & Painters** (O'Reilly, 2004). He has a Bachelor's degree in Philosophy from Cornell and his Masters and a PhD in Computer Science from Harvard, he has 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 agree. I was curious about his PhD dissertation, so I tracked it down. It's titled: "*The State of a Program and Its Uses (1990)*". That's wonderfully mystical. I read the Abstract. As you might expect, it's some seriously nice pure computer science thinking.

So I poking around a bit more, I looked at Paul's Twitter postings. A recent post from Saturday was:

> ***In office hours today*** [and I should mention that, as we'll see, his use of the term "office hours" is important to him] *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:

> *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.*

And looking a bit further back, on August 1st, Paul Tweeted:

> *The hardest people for founders to hire are so called C-level executives, because these people are the best fakers in the world. Even the best founders make absolutely disastrous mistakes hiring these people. It happens far more often than anyone realizes, because neither party wants to talk about it. So after nearly destroying one company, the exec cheerily goes off to their next opportunity.*

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 one and a half million current followers. I'm unsure why I find this guy so fascinating. That really never happens. But he's got something.

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. That might be what's hooking me. He made money early in the run up of the Internet and 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 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. But I made the decision to share this, as this week's topic, even before I knew that.

Paul explicitly gives 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. I've done that 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": http://www.paulgraham.com/makersschedule.html

Maker's Schedule, Manager's Schedule

By Paul Graham

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 one meeting can sometimes affect a whole day. A meeting commonly blows at least half a day, by breaking up a morning or 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, 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? 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.

Our case is an unusual one. Nearly all investors, including all VCs I know, operate on the manager's schedule. But Y Combinator runs on the maker's schedule. Rtm 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 till about 3 am every day, because at night no one could interrupt me. Then I'd sleep till about 11 am, 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 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 that they understand the cost.

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 "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.

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. This 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 — and I want to be. Pretty much most of the time I just want to be left alone to work. If I'd had the wisdom back when GRC was a caldron of chaos, I should have worked at home in seclusion 4 days a week and held office hours, as Paul does, on Fridays. On those Fridays, I would have resigned myself in advance to not getting any code written that day... and perhaps I would have been both more available and less miserable.

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

Paul's "Articles" page (http://www.paulgraham.com/articles.html) lists this and 211 other short essays. I've only poked around them a bit, but I find something about them to be quite compelling. I didn't want to stop reading them, but I had this podcast to finish. The Sci-Fi book I've been treading water with, while waiting for Ryk Brown's next installment of his Frontier's Saga, has been disappointing. So instead, I will likely read all of Paul's 212 essays. We've already just read read one. At the top of his essay's page, Paul writes:

> 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*".

So this week I leave you all in Paul Graham's very capable hands.