A Snowflake's Chance

Description: How can content delivery networks be used safely? What do we learn from the ransomware attack that affected 15,000 auto dealers? Guess who uses an Entrust certificate and when it expires? How worried should we be about Polyfill.io attack aftermath? Whose side is Microsoft really on? Let's look at their history. How is GRC's new weekly Security Now! mailing going? And what about feedback? And, finally, the company named "Snowflake" was the epicenter of what has now become the largest series of corporate data breaches in history (and that's saying something). Naturally there's been a lot of finger-pointing. So who's saying what, and what appears to be most likely?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-983.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-983-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have some really interesting things to talk about. An update on the Polyfill.io attack. Can content delivery networks be used safely? That ransomware attack against 15,000 auto dealers, what have they learned there? And you won't believe who still uses Entrust for their certificates. I'll give you a hint, it's got a .gov domain. Finally, we'll talk about the Snowflake breach. Steve's not sure he believes the stories. He's going to get to the bottom of this one. All coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 983, recorded Tuesday, July 16th, 2024: A Snowflake's Chance.

It's time for Security Now!, the show you wait all week for, looking to collect the little tidbits in the back of your brain, at least I do, going, oh, I wonder what Steve's going to say about that. Now here's your chance. Steve Gibson is here, our master of ceremonies. Hello, Steve.

Steve Gibson: Yo, Leo.

Leo: Good to see you.

Steve: Great to be with you.

Leo: Yes.

Steve: One of the final episodes we're recording with you in the Eastside Studio.

Leo: In the old studio, yeah.

Steve: That's right.

Leo: There'll be, let's see, there'll be two, three more because we're leaving August 8th. So August 7th will be the last episode of Security Now! from the Eastside Studio, which has been - it's been a good run.

Steve: So that will be after This Week in Google on Wednesday afternoon?

Leo: Yes, yes.

Steve: Will be the last recording.

Leo: Then that's when I'm going to...

Steve: That will give you a few days, Thursday, Friday, and Saturday...

Leo: To spread the accelerant, stuff the newspapers in the various crevices, light the flame. No, we're not going to do that. We're not going to burn it to the ground.

Steve: Okay. So we've got a lot of fun stuff to talk about, and some - sort of some interesting takeaways, I think. Today's podcast number 983 for this July 16th is titled "A Snowflake's Chance." This of course named after the firm, and I don't know why they named themselves Snowflake. That just sounds like a flaky...

Leo: Exactly, yeah.

Steve: You know, it's not a good name.

Leo: Now that you mention it.

Steve: Yeah, I don't know that I want to store all of my customer data at Snowflake.

Leo: That's the one that has a chance in hell. I mean, that's not a good name.

Steve: No.

Leo: Yes.

Steve: No. And at least 350 companies are now very sorry that they did store all their data there. And, you know, Ticketmaster, of course famously now AT&T, with 110,000 of their customers' mobile phone metadata, and the people that use the AT&T system and which AT&T resells to. Anyway, big disaster there. Some interesting takeaways from that.

But we have a lot of other stuff to talk about. The discussion last week of the Polyfill.io attack caused people to say, can content delivery networks be used safely? Because of course the problem was that Polyfill.io the domain got purchased by a now-known-to-be-malicious Chinese company. So what's to prevent that from happening elsewhere? Also, what do we learn from the ransomware attack that took 15,000 auto dealers down a couple weeks ago? And interestingly, I heard from three of our listeners who were directly impacted by this.

Leo: Oh.

Steve: I mean, so this thing was widespread. Also, guess who uses an Entrust certificate, and when it expires? We're going to look at that. Also...

Leo: Oh, that's interesting. Can't wait to hear that one.

Steve: Don't fool with the IRS.gov until you catch up on...

Leo: Oh, oh.

Steve: Also, how worried should we be about the Polyfill.io attack aftermath? That is, do we have to reformat our hard drives, or what? And a listener really brought up an interesting view that I'm going to share about whose side Microsoft is really on by taking a look at their history. I'm going to comment about how GRC's new Security Now! mailing is going. And then, as I said, we're finally going to take a look at this company named, who knows why, Snowflake, which was the epicenter of what has now become the largest series of corporate data breaches in history. And as we know, unfortunately, that's saying something.

Leo: Geez.

Steve: So there's been a lot of finger-pointing, who's saying what, what appears to be most likely. And always we have a fun, but puzzling, Picture of the Week. I understand what's going on. I've already had feedback from those who received Security Now!'s email a couple hours ago saying, what? Anyway, we'll explain it.

Leo: I haven't seen it yet. We will see it together for the first time.

Steve: I think everyone is going to like - they're going to find these next two hours have been well spent.

Leo: I think that's the case, as always, Mr. G. And we have Adam watching on Facebook today. Thank you, Adam. It's very nice to have you. You don't know this, but we are streaming - maybe you do know this, Steve. I don't know. We're streaming now everywhere.

Steve: But we're using Zoom.

Leo: We are. But we have a service called Restream that we've started using that's going to be - we used it a couple of weeks ago. Remember I was pulling up chat, and it was distracting you?

Steve: Ah, so you could either use Restream for the conferencing or use Zoom as the frontend and then Restream on the backend.

Leo: That's exactly what we're doing. We're going to have Zoom ISO in the frontend. We're going to have Ecamm, so the technical director, our producers will also be switching the show. I won't have to do that anymore, which is interesting. Actually we'll be doing less, I guess.

Steve: Good, because those Restream fades were not good.

Leo: No. These will be much nicer. But we do have Restream taking our video and putting it everywhere. So it's still YouTube, YouTube.com/twit/live. But it's now Twitch.tv/twit. It's on Facebook. It's on LinkedIn. It's on X.com. All over the place. So Adam is watching on Facebook, and I guess hasn't seen the video or doesn't remember. But he said, "What's that Speak & Spell right behind Steve there?" But there's a story with that; right?

Steve: Yup. I was involved in its creation.

Leo: Isn't that amazing. So that's one of Steve's products, I guess you could say.

Steve: Well, no. Linear predictive coding speech synthesis was the technology used.

Leo: Wow.

Steve: And that's part of what happened when I was at the Stanford AI Lab.

Leo: It's kind of amazing. I mean, here we are, what was that, 40, 50 years ago? Fifty, I guess.

Steve: Yeah, it was '73 was when I graduated high school, and I was at the AI Lab for the couple years before that.

Leo: It's kind of amazing. And look where AI is now. I mean, yeah, the sound of the voice synthesis in the Speak & Spell versus what ElevenLabs is doing now with celebrity voices, I mean, it's incredible. We have lived in interesting times, Mr. Gibson.

Steve: One of our listeners, one of our female listeners took offense to my little rant last week about man in the middle and needing to call it, what was it, not attacker in the middle, adversary in the middle. And she fed what I said into some generative AI, asking it, how could women find this offensive? And it was astonishing. I mean, I really - it put me in my place. I thought, well, maybe you just need to go, we need to turn you out to pasture because you're obviously, you know, you need to have a serious sit-down with human resources.

Leo: Or at least with ChatGPT. Gemini wants a word with you, Mr. Gibson.

Steve: Uh-oh.

Leo: I have the Picture of the Week right here on my laptop, and I am ready to pull it up. Do you want any prelude?

Steve: So I gave this one the title, "Does anyone wonder how to lock this bathroom door?"

Leo: Oh, my imagination is reeling. Let us look together. Apparently, no one knows how to lock this bathroom door. There are one, two, three, four, five, six signs, all of which say, "Do not turn. Push to lock. Please."

Steve: Five of the six are in a san serif font of varying sizes, apparently having been incrementally added to the door. Five of them are on the door. One is over on the wall. Somebody came - the one that is the serif font, which reads "Simply push to lock," where those four words fill the entire page, someone came along and gave it some extra underlining.

Leo: Underlining, yeah.

Steve: In felt tip marker.

Leo: Simply push.

Steve: So we have "Just push to lock. Do not turn" with three exclamation points. "Push to lock, do not turn," oh, that one was actually - that looks like it's pretty much the same - no, no, no.

Leo: That's replicated.

Steve: No. The first one says "Just push to lock." The second one says "Push to lock." So apparently that wasn't sufficient, so they added this "Just push to lock."

Leo: Don't turn it.

Steve: Now, we actually have a paragraph down at the bottom. "If you just push the button straight in, without turning, the door is locked."

Leo: Now, this makes me wonder, what happens if you turn it? In fact, honestly, being the rebel that I am, looking at these signs, I would turn it.

Steve: Apparently people do. I know what's going on here.

Leo: Oh, good.

Steve: If you push in - so we should explain, it's got sort of an L-shaped handle coming off to one side. And if you push it down, the door will open, and you can leave.

Leo: Yeah.

Steve: But the button is actually sort of a thumbscrew...

Leo: It looks like you should turn it. It's got a little affordance for turning it.

Steve: Oh, it wants you - it is saying "Turn me."

Leo: Turn me, yes.

Steve: Everything about this. So you push it in, and it stays in. Then when you push the handle down, it pops out.

Leo: Right.

Steve: But if you push it in and turn it, it locks the button in.

Leo: It locks you out after you go through the door.

Steve: No, no, no. Well, yes, exactly. So what people are doing is, because this was a poorly chosen handle for a bathroom, they're pushing it in and turning it, thinking, okay, I don't want anyone coming in on me.

Leo: Yeah.

Steve: When I'm in the middle of doing my business.

Leo: Right.

Steve: But then they leave. And because they twisted, the door locks behind them, and nobody can ever get in again. Now, I don't know if residential home door jamb locks still have this. But Leo, I know that when you and I were young, if you looked at the door jamb on the front door of your home, there would often be two buttons there.

Leo: Yes.

Steve: And those two buttons would lock or unlock...

Leo: Right.

Steve: ...the thumb lever on the outside of the door, the point being that it would keep the door locked from the outside, whether or not you locked it from the inside. It's exactly this technology.

Leo: I see.

Steve: Unfortunately, they made this much too easy to use.

Leo: And they put the wrong affordance on it because they put a little turning thing on it, which you shouldn't have. This, by the way, if you've not read this book, the great Don Norman's "Design of Everyday Things." And the cover tells you all that you need to know. It's a teapot with a handle on the same side as the spout. Which obviously is poor design. He talks about this all the time, doors that invite you to push it because they have a push bar, or worse, have a handle for pulling, but do the opposite. It's very common; right? So they have to put a sign up that says "Push, don't pull." Because you pull it and nothing happens, and you look like an idiot. So this is very common, and he says, "Don't blame yourself. This is just poor design."

Steve: Well, and what this company, whoever they are, should do is just hire a locksmith or a doorknob person to come out.

Leo: Fix it.

Steve: And put a button here. I mean, because you can buy the same handle with a button, and then you don't have this problem.

Leo: Yup. And that's the other thing, Dave Redekop's saying...

Steve: And clearly, no amount of signage is going to solve this.

Leo: No. If you have - that's, you know, a little hint. If you have eight or nine signs on the door explaining in the same way how to do it, the message isn't getting through. It's just poor design.

Steve: Yeah. And presumably somebody has to go when they come into this bathroom. So they're not taking any time to, like, read the dictionary of proper knob operation. There is some hurry.

Leo: Well, Dave also - Dave points out that the other thing they may do is push it and then test the handle and inadvertently unlock it, which could also lead to embarrassment. So...

Steve: Bad design.

Leo: We need better design, better design, exactly.

Steve: Okay. So using content delivery networks safely. Looking back upon last week's "Polyfill.io Attack" topic, I can imagine that I may have come off as being very anti-third party when it comes to sourcing potentially dangerous content, such as code libraries, from third parties, you know, such as high-performance content delivery networks, CDNs. It was never my intent to rain on the idea, you know, the concept of CDNs in general for this purpose, because the web's designers have made ample provisions for safely pulling code into web pages from remote sites. And a number of our astute listeners sent me notes asking variations of "Uhhh, Steve, did you perhaps forget about asset integrity pinning?"

Actually, no, I didn't. But those questions also raised a very good point. So rather than answering each of those notes separately, and since it's a terrific topic for this podcast to cover in the wake of the Polyfill.io news, I wanted to talk about how third-party content can be delivered safely, and why the Polyfill.io facility was never able to take advantage of that.

Okay. So the formal name for the facility is "Sub-Resource Integrity," abbreviated SRI, where the concept and implementation could hardly be clearer, cleaner, and simpler, you know, as the best things are. This is a win. The same HTML <script> tag that contains the URL of some remote third-party code or stylesheet, because it could be used for link tags also, where the URL is what the browser is being asked to remotely load, can also, optionally, contain another name/value pair, specifically integrity= and then a big quoted string.

The format is the word "integrity," followed by an equal sign, then any one of the prefixes "sha256," "sha384," or "sha512." And our astute listeners will already be going, ah, I know what's coming. What's coming is a dash, followed by the specified hash of the expected URL resource which has been hashed under the specified hash, then encoded from its binary into Base64 ASCII.

Okay. So here's what all that means. When a web designer wishes to pull some remote resource from a remote content delivery network, or really from anywhere, where they do not directly control that resource, they want to be absolutely certain that the resource they want has not been changed from what they expect to receive. So they first go to the SRI hash generator site, very handy, www.srihash.org. And this is formally specified like at the W3 Consortium. Mozilla talks about it. This is sort of like the reference site for generating these resource protection hashes. Again, www.srihash.org.

Or you could OpenSSL or any other utility that can create Base64-encoded hashes. The srihash.org site is handy since, when given a URL, it will fetch the resource for you, perform the hashing, perform the Base64 encoding, and return the snippet of code tag which already is set up to drop into your own web page to perform all of the proper matching. It defaults to sha384, sort of the one of medium strength. But really sha256 is plenty strong also, creates a somewhat shorter hash. But, you know, these days web pages have become so out of control, no one cares.

Anyway, so our designer goes there, gets the hash, drops the URL of the jQuery library they wish to use into srihash.org to receive the hash. Then they add this hash, along with that "integrity" keyword, into the jQuery fetching script tag in their web pages, and they are henceforth protected from any modification of that code.

When their page is delivered to a user's web browser, the browser reads the page's HTML, sees the <script> or <link> tags, and fetches the resource referred to by the URL. But because that <script> or <link> tag also includes an "integrity" argument, before the browser does anything with the freshly downloaded resource, it takes its own hash of what it just downloaded. Base64 encodes that and compares the result with the hash that follows the "integrity" keyword. And only if the hatches match will the browser allow that code to enter the browser's inner sanctum to be trusted and used.

So what all of this does is very nicely and cleanly allow web designers to protect their sites' users from both inadvertent or deliberate alteration of the resource that they're requesting. Also note that since code libraries are constantly evolving - jQuery is currently at, for example, v3.7.1. I noted that the jQuery my email system is using, I think it's 3.6.0. So jQuery is a moving target. Because of that, it's also necessary for the jQuery or whatever library specification to indicate the exact version that is being desired so that the hash will match. Since the CDNs will always continue to offer all older releases, a site will continue to use that version, the one that's known to work, until its designer changes the version number in the URL and obtains that newer release's matching hash to add to the invocation tag in the HTML.

Okay. So we now see how it's completely possible to safely obtain potentially dangerous script code from any other service that the designer does not control. All of the advantages a CDN has to offer, like nearby points-of-presence so you get super-fast content delivery of potentially large content without delay, that can all be used without any risk at all, by specifying the hash that you expect that content that you receive to have. And the browser itself will just say nope, and will not use it

But as I said earlier, unfortunately, this very slick protection was not available to users of Polyfill.io. I touched on this briefly last week; but since it wasn't our focus, I didn't elaborate or highlight its significance. So just now when I was putting the show notes together, I used our trusty Web Archive's Wayback Machine to show the Polyfill.io website home page. They were clearly - they, the Polyfill.io people, when they originally created this were clearly very proud of what they'd created. And they were a little tricky.

The home page of the site says: "Just the polyfills you need for your site, tailored to each browser." And then they said "Copy the code to unleash the magic." And what we see is a script tag with a, you know, src= and then a URL. Which doesn't have any specific

subversion numbering. It says v2, but otherwise no. Polyfill.min.js. Then it says: "Polyfill.io reads the User-Agent header of each request and returns polyfills that are suitable for the requesting browser." They said: "Tailor the response based on the features you're using in your app, and see our live examples to get started quickly." In other words, they are customizing what they return, depending upon the browser the user is using. No version numbers.

So the way the Polyfill.io site always worked was that it generated and delivered custom polyfill JavaScript code specifically tuned to the make, model, and version of the web browser each individual user was using. And Leo, I'm going to need you in about a minute. This always made it actively hostile to...

Leo: I won't leave.

Steve: ...to the web subresource integrity system, which prevented any of the Polyfill.io's great many users from supplying a hash code which they would - because they would receive entirely different code each time. Anyway, that said, our listeners were 100% correct to point out the power and value of subresource integrity protection. It's been universally supported by every web browser for many years, so it's something that all web designers whose web pages are pulling code, which should never change without notice from any third party, should be adding to their bag of tricks.

So I did want to just bring it to everyone's attention, that very cool website that I mentioned. Srihash.org will give you the hashes and the code to drop in to any remote resources you're pulling. So it is entirely possible to protect yourself. So again, I thank our listeners for bringing my attention back to that. I wanted to make sure that I explained. And it's absolutely possible to safely pull remote content from CDNs. Not from Polyfill.io because it was always delivering customized code.

Leo: Yeah.

Steve: Otherwise, yes.

Leo: Interesting. Yeah, I mean, how often do we install stuff that says, oh, just copy and paste this line that will download and install software. And I, you know, there's a program I put on every Mac called Homebrew. And that's how you're supposed to install it. You know, I mean, you could look at the bash script, I guess. But it's just - it's very nerve-wracking. You nailed it, though, last week on talking about the Polyfill. In read, I wanted to mention that in our TWiT Forums at TWiT.community, Pseudorandom Noise said: "Another great episode. And as a web developer, it was nice to hear how incredibly correct Steve was and how well he understands the subject matter." So I just thought I'd pass that along. People really appreciate your deeper understanding of this stuff.

Steve: Thank you.

Leo: And I guess because you're a coder, you know, you know about libraries, and you know how tempting they are to load. And as it turns out, thanks to supply chain attacks, how risky.

Steve: Well, yeah. And, you know, I'm a bit of a web developer myself. I don't have, as everyone knows, a fancy website that's taking advantage of all kinds of...

Leo: You're not using React or anything. But, you know, it's pretty good. It's nice. It looks good.

Steve: I'm not even using JavaScript.

Leo: It's all text. All HTML.

Steve: All of the - that's right. Even the GRC's web, I mean, our website menus are just using CSS with no scripting in order to perform their magic.

Leo: Nice. So that's all you need, frankly. That's good enough, yeah.

Steve: It works. But let's take a break because the next chunk is going to be a big one, about this massive CDK Global ransomware attack that impacted three of our listeners.

Leo: Oh, dear. And now another breach. Steve?

Steve: Ah, yes. Toward the end of June I heard from three of our listeners whose lives have been affected by a recent major attack on a very large automotive dealer network. On June 21st I received two notes from our listeners. First one said: "Hello, Mr. Gibson. My name is Shawn, and I'm an automotive technician at a GM dealership and have been listening since about 2016."

Leo: Oh, I know what he's going to talk about. Oh, I know what he's going to talk - this hit my dealer, too.

Steve: Yup. Yup.

Leo: Okay. Okay.

Steve: He says: "I love when your world of security crosses over to my world of automotive. My dealership, as well as thousands of others" - yes, 15,000 of others - "is affected by the CDK cyberattack that happened yesterday. When the details come out, I would love to hear your take on it. This is the first time a cyberattack has had a direct effect on me." He said: "(We get paid by what we do, and this is slowing everything down as we have to go back to manual ROs and quotes, lowering my booked hours)."

Leo: Oh, that's too bad.

Steve: "Thanks. Shawn."

Leo: Yikes.

Steve: And then we actually heard from, apparently, an owner of a dealership. On the same day, on June 21st, I received: "Steve, thanks for all the years of podcasts. I've been a listener from the beginning and a watcher from the Tech TV days. I hope to hear some coverage of the CDK Global incident. Sales of auto repair parts from the dealer side of the industry have come to a screeching halt as they're unable to create invoices nor tell us our cost for a part. I was told today from one dealer that they hope to be able to sell me parts next week with some form of paper invoice." Paper, imagine that. "The delivery box truck that stopped was almost empty today." And he said: "I only got my parts today as they had already been ordered and invoiced prior to the issue. Thanks. Signed Alan Alberg of Alberg Auto."

And then finally: "Hi, Steve. I just found out about this this evening. Our son and daughter-in-law are both remote workers for a dealership network that has been brought to a standstill by this cyberattack. USA Today is reporting 15,000 dealerships across the United States are affected and may not be back online until the end of the month. Color me cynical, but I'm fond of the saying, 'There is no cloud. You are just someone else's computer.' Usually that other computer is better secured than your own. But as you so frequently say, 'It's not a matter of IF, but WHEN.' I appreciate the work you and Leo put into the podcast each week. Best regards, Richard in Clemmons, North Carolina."

Okay. So what's the scoop on this? We have a situation where 15,000 operating dealerships were dependent upon a single MSP, a managed service provider, also sometimes referred to as SaaS, or delivering, offering SaaS, Software as a Service. The dealerships were dependent up on this for all of their, I guess we still call it "paperwork" processing, though it's virtual paper. We'll be talking more about Software as a Service when we get into today's discussion of the Snowflake disaster. But in this case a Russia-based drive-encrypting ransomware cyberattack took down hard the entire network of 15,000 auto dealerships which needed that network to operate.

I found a terrific piece posted on Medium by someone who's been in the auto industry and writing about it for some time. Speaking from her long experience, Kathi's headline is: "The CDK Cyber Attack Recovery Will Fall Squarely on the Accounting Office." She writes: "During my first years in the car business, I wore a lot of hats in each job position I had. The one thing I learned early is that the accounting office staff are often the cleanup crew when several types of problems arise. There are still systems and procedure hiccups that happen today, but thanks to technology and automation they're fewer in number. Then came the CDK cyberattack.

"This CDK cyberattack is on a whole different level. This breach is a very different type of problem. But in the end, when things begin to settle, which may take months, it will be the accounting office who will be tasked to gather the thousands of dealership puzzle pieces from sales, service, and parts, and methodically match them up together to form some semblance of financial order.

"The 'End of the Month' is here." So she was writing this at the end of last month, June. She said: "New car dealerships are required to produce a monthly financial statement as mandated by the manufacturer and certain lenders. It's unclear as of this writing if a June financial statement will be available. I would say the chances are slim.

"So why did the CDK cyberattack happen? There was once a company called ADP Dealer Services who were a great DMS provider. DMS is Dealer Management System," which is

the generic term in the industry. "They got rolled into a company called Cobalt that sold mostly digital marketing services. Then all of that got rolled into CDK Global, and with that came" - yeah, wait for it - "private equity investments." Now, this is her speaking, not me. And I'll address this a little bit later, in a second.

She said: "The first thing to get cut when private equity rolls through the front door is 'cost centers,' and Infosec (aka: Information Security) is viewed as a cost center. The main people who defend the gates of the village (the company) from the barbarians (the hackers) are the first sent off to exile. When there's a ransomware attack, it's revealed with clockwork-like precision that no one has tested the backups for six months, and half the legacy systems cannot be resuscitated.

"As a cybersecurity expert told me last week, a few days after the attack happened: 'It's been at least two days since the ransomware attack with no fix in sight,' which tells me," he said, "a few things on this list have to be true. A, they have no backups. Or B, if they do have backups, they're outdated or never tested, which is effectively the same as having no backups. C, no one knows how to restore the backups. D, there's no disaster recovery plan; or if it exists, it's outdated to the point of uselessness. E, multiple single points of failure are baked into the infrastructure. Or finally, F, they have no idea how compromised they are."

So she says: "I'm very angry about how ADP Dealer Services, once a great company, has been raped and pillaged by private equity. The real pain is suffered by the rank and file at the dealerships, who still have to care for customers and sell to make a paycheck. According to recent reporting, CDK will be paying the tens of millions of dollars in ransom." And I've got some follow-up reporting on that I'll share in a second. So I'll just note that information security doesn't seem like such a waste of money now, at this moment, does it.

So she says: "How did the CDK cyberattack happen?" She says: "CDK is an ancient program. Not a lot has been done to upgrade the original version for decades. This is standard operating procedure when companies or private equity buy legacy companies. Innovation is not the goal. They slap on a new paint job or buff out the dents, and package it as the 'new improved version' that is always more expensive, but 'worth the investment.' Ask any dealer how they feel about CDK and other DMS fees these days. These corporate raiders' goal is to cut costs at all costs. And in this debacle, it's clear they stripped the car for parts and left the data vulnerable to cyber criminals."

Now I'll interrupt to just say, as we know, it's very difficult to completely protect any large organization from intrusion. But her earlier point about recovery is unassailable. Any organization today whose survival would be threatened by a significant protracted network outage should certainly arrange to get back on the air after any attack.

Anyway, Kathi continues: "Theoretically, a mature Dealer Management System provider should be able to lose any single critical part of their core business and be able to restore functionality within 24 hours, barring a massive natural disaster or personnel losses. Instead, they have no backups, no redundancy, no separate servers, and no siloed databases which, when lost, are a pain to retrieve, but at least it's only one silo and not the entire client roster of 15,000 locations.

"How does a dealership restore their records once the breach is contained? Once CDK pays the ransom, it may take weeks or even months to get all the data in order after they receive the keys to the ransomware," she writes. "The database will likely have holes in it that will add to the arduous restoration process." She says: "There's been a lot of talk online about just getting a new DMS vendor. While that seems like a good solution, the problem is that your data is being held hostage by whoever attacked CDK. Without the data, you have nothing to convert to the new DMS. But the idea of other

DMS solutions is a good one that should be explored once the dealership's CDK records are restored.

"When the dealership comes back online, that's when the fun starts for the Accounting Office. During the outage, all employees continue to serve customers to the best of their ability, using manual documents and a patchwork of software support."

Leo: A bunch of paperclips.

Steve: Exactly. Like, well, like remember when we're in a restaurant, and their credit card processing goes out.

Leo: Yes,

Steve: And it's like, uh, whoops. Yeah. So she says: "When operations is functional again, all the business they produced new and used car sales, service, parts, internals, warranty anything that happened during the downtime will need to be assembled and manually input into the system. It could take a few weeks or a few months to match everything up, and it will be a lot of work just to get back to 'normal.' Organization is key. If it's a busier store think 150-plus cars per month or over $500K in monthly service labor it will take a considerable amount of time to input due to the sheer volume of transactions. Vehicle inventories new cars, used cars will need to be counted to verify every unit's whereabouts. Parts inventory should also be verified unless the store had some kind of redundant system that kept track of it during the outage. Untracked inventories are ripe for theft.

"If all the manual input goes well, and I do mean 'if,' she writes, then all entries should land in their respective general ledger accounts. Schedules and other general ledger reports should be run to determine what it all actually looks like, and to make sure all the monies that were collected are posted to their respective accounts. One surefire place to start is bank reconciliation. If you can balance your books to your bank, you'll have a roadmap to a decent amount of checks and balances. It will not be pretty. But with the always-present perseverance of dealership accounting office staff" - sounds like she was once a dealership accountant.

Leo: Oh, this is definitely a post-traumatic stress syndrome from somebody who's been on that front line, for sure, for sure.

Steve: Exactly. She says: "It will ultimately come together." And she says: "I'm just so appalled that this event happened. When I first heard about it, I said to my colleagues: 'In what universe is it okay to manage data in such an irresponsible way?'" And I'll have something to say about that, too, in a second. She said: "Most dealership employees have never had to perform their job without the use of technology. It's a strong reminder that technology is only a tool for efficiency, and it's only as good as its infrastructure and established crisis protocols.

"There will be lawsuits, of course. The only question is how many and from whom. Certainly I would expect claims against CDK from" - and she names three: "Dealers for impeding commerce and negligence in data loss, among other things; consumers for the massive data breach of extremely sensitive information; and employees for data privacy and lost compensation." She says: "Now is a good time for dealers to contact their Cyber

Liability Policy carrier. Check to see if you have Contingent Business Interruption coverage and put the carrier on notice. No need to file a claim just yet, but it's worth having a conversation to know if you're covered and for how much."

Okay. So that's Kathi's take. In subsequent reporting, CNN Business reported, under their headline "How did the auto dealer outage end? CDK almost certainly paid a $25 million ransom." So CNN wrote...

Leo: Oh, my god.

Steve: Yup, 25 million to get back online. CNN said: "CDK Global, a software firm serving car dealerships across the U.S. that was roiled by a cyberattack last month, appears to have paid a $25 million ransom to the hackers, multiple sources familiar with the matter told CNN. The company has declined to discuss the matter. Pinpointing exactly who sends a cryptocurrency payment can be complicated by the relative anonymity that some crypto services offer. But data on the blockchain that underpins cryptocurrency payments also tells its own story.

"On June 21st, about 387 bitcoin then the equivalent of roughly $25 million was sent to a cryptocurrency account controlled by hackers affiliated with a type of ransomware called BlackSuit. A week after the payment was made, CDK said that it was bringing car dealers back online to its software platform." They write: "Cryptocurrency allows for the exchange of digital assets outside of the traditional banking system, but a record of those transactions is accessible on the blockchain. Three sources closely tracking the incident confirmed that a roughly $25 million payment had been made to BlackSuit affiliates, and that CDK was very likely the source of that payment. Those sources spoke on the condition of anonymity because of the sensitive nature of the investigation. The cryptocurrency account that sent the ransom payment is affiliated with a firm that helps victims respond to ransom attacks, one of the sources said, declining to identify the firm."

Okay. So will the payment of that $25 million affect CDK's behavior going forward? Who knows? The greatest cost is likely their reputation damage. We've previously seen the consequences of MSP's - managed service providers - being penetrated to allow malicious attacks against their clients thanks to the MSP's access into those clients' networks. But that's not what happened here. The problem here was that 15,000 auto dealers had come to so depend upon the networked services provided by their massive MSP - and I'm sure that was both at the MSP's urging and the dealer's willingness to avoid redundant work - that when that MSP was taken down by a ransomware attack, the second-order consequences were so widespread that at least three listeners of this podcast were directly affected and wrote to me.

Whether or not this was a consequence of profiteering by private equity owners, who stripped the organization of what they felt were excessive cost centers, is irrelevant here. We've certainly seen many organizations attacked with devastating consequences when their owners were fully invested in their company's success and infrastructure security. And as we know, mistakes happen. Could profiteering ownership have been a contributing factor? Okay, sure, maybe. But we would need to have much more information about CDK Global's history to render any judgment about that. The point Kathi made in her article about there needing to be some explanation for the fact that CDK Global was unable to recover immediately without paying a multi-million dollar ransom, that was certainly a good one. But we have no idea what's going on behind the scenes and whether, you know, and what one way or another was the case.

And to my mind that's really beside the point. What I think we have here is another consequence of a theme we saw last week with Polyfill.io, where so many websites were pulling unverifiable code from a central source. This is another example, and just wait till we look at the Snowflake disaster in a minute. One way to describe all of these widely different problems would be as the danger of the promise of a free lunch. Or stated another way: "It's very rare that you get something for nothing."

Remember that xkcd cartoon we showed last week, where a massive construction of blocks was ultimately resting on an endangered twig? In the case of the CDK Global MSP outage, 15,000 auto dealerships had become dependent upon this single service provider for virtually all of their daily operations. And it's entirely human for this to happen over time if CDK's service had been so reliably delivered for so long that the maintenance of any "backup plan" in the event of a CDK service outage seemed entirely redundant. For all we know, there were such plans in place 15 years ago.

But staff changed, people who knew how to fall back to a manual system retired and left the dealerships, and new hires were only trained on and knew how to use the automated system. You know, "Just press this button and follow the onscreen prompts." So what gradually grew over time was a deepening dependence upon this miraculous new system that had, after all, demonstrated to be dependable enough to be depended upon, right up until the day its plug got pulled. And without it, a massive network of auto dealerships were marooned.

Kathi was correct in her prediction that class action lawsuits would be filed against CDK. Some already have been. And I think that's unfortunate because the whole truth is, this sort of free lunch failure is what comes with the territory. Class action lawsuits after the fact, when the free lunch needs to be paid for, is just sour grapes. Having tasted and grown accustomed to the power of the service provided by CDK, it would be safe to predict that not a single dealer is going to return to a manual in-house operation. Was the pain that great? No. Not nearly enough. Might some switch to an alternative provider? I'd bet that even that is rare.

Everyone is breathing a huge sigh of relief, with the network and the automation that it provides coming back up, and business resuming as normal. CDK's CEO has apologized. He's promised to improve their cybersecurity posture and has even offered some financial restitution to their 15,000 dealerships for the loss of sales and service revenue that they suffered. And life goes on.

So the message I'd like to take from this perfect example of what can go wrong is that, in the final analysis, it's all worth it. I don't take the opportunity to remind us of that often enough. You know, we're only doing all of this cyber stuff because it really does make sense. It really is phenomenally powerful. It really is improving people's lives. Sure, there's a "two steps forward, one step backward" sense. And that faltering backward step can be painful. But the net effect is still one step forward.

This still doesn't mean that a truly massive catastrophe is not possible. From all the evidence we continually see, we can feel the very real possibility of that in our guts. And following from my analogy last week of hoping for minor earthquake tremors, the hope is that other competing DMS (Dealer Management System) providers are looking at what just happened at CDK and shuddering, while suddenly feeling better about the size of their own information security budgetary line item. And they ask their IT staff with renewed attention whether they are safe from the same thing happening to them; and, if not, what more do they need to do?

Thanks to the network effects of this event, a great deal of press coverage and attention was given to this. So let's hope that some lessons were learned to better prepare other similar organizations to respond if it should happen to them.

Leo: But what lessons can be learned? No dealership is going to turn to a tool that they create themselves, nor would that solve the problem. CDK I'm sure says, well, we're going to make sure we're more secure. But are they, and will they, and can they?

Steve: I don't think it matters, Leo. I think that dealerships are saving so much by using, by basically subcontracting out, I mean, the very fact that they were completely crippled demonstrates how much of the work of...

Leo: How useful it is, yeah.

Steve: Yes, how much of their work CDK was successfully automating.

Leo: Right.

Steve: So yes, this was not good. They were inconvenienced. You know, like the restaurant whose computers go down, and they have to take orders by hand and process their customers' credit slips. It's inconvenient. But do they stay on paper afterwards? You know, they recover, and life goes on.

Leo: Yeah.

Steve: And that's really the message I wanted to convey here is, yes, this was big. This was awful. But ultimately this was a minor earthquake tremor. This is what we want. We want to remind people these things can fail. And failure, we would like it not to happen, but it does. And but even so, when you step back from it, all the dealers are going to stay with CDK. They're going to say, well, the CEO says they're going to improve their security. And look, we got a check that didn't begin to cover our losses, but at least it's something. And besides, it would be far too painful to have to train our staff how to do this themselves if we didn't have automation. And we don't really want to move to a different DMS provider because all of our stuff is here. So look, it's working again. The lights are on. Let's just move forward.

Leo: Yeah, and I don't even know what you could do. It's, I mean, it's almost like saying, well, don't use computers. You know, go back to that paper system. That never crashes. Nobody's going to do that.

Steve: Right. Now, I mean, I guess the only thing you could do, if you really cared, would be to go to the expense of running your own system, and not using a Managed Service Provider. However, there may very well be significant advantages from being tied into CDK. There was some stuff Kathi talked about that was so deep in the weeds that I didn't include it. But it was - it had to do with CDK's stature as a preferred vendor to auto manufacturers. And that gave them some privileged status which all of those 15,000 dealers inherited as a consequence of using their dealer management system. So they're actually, you know, there actually is benefit that an independent dealer being entirely independent, even from automation, would not have.

Leo: Yeah. And you can't even really fault CDK. I mean, I wish they - I almost want to say it's too bad they paid the ransom. But there was no alternative.

Steve: No.

Leo: That was the cheapest way of them getting those desperate dealers back online.

Steve: Yes. And we can fault them for not having, I mean, they're probably faulting themselves for not having the infotech, the information technology security, to deal with this. They've been around a long time, for decades, before ransomware became a problem.

Leo: Right.

Steve: I'll bet you this is pure inertia. They just hoped it would never happen to them. And they were using backup technology and security that was a decade old.

Leo: Yeah.

Steve: So when this thing took them out and encrypted their servers, they were...

Leo: They were unprepared.

Steve: To use the phrase that's common these days, screwed.

Leo: All right. We'll have more in a bit. When I ask, when I say, well, what can you do, there are things you can do. Just listen to our sponsors. I mean, this is specifically why advertisers come to Security Now!, because they're trying to reach out to companies like CDK, saying before it's too late.

Steve: So a listener, Knox North, he said: "I listened to the Entrust story with interest, even though professionally I use DigiCert. I figured I'd never encounter Entrust. But I went to https://irs.gov, and guess who issued their cert?"

Leo: No.

Steve: Now, Knox's observation made me curious, so I went over to the IRS.gov website to see for myself. First of all, sure enough, the IRS has been purchasing its websites' TLS certificates from Entrust. Presumably that will end. But what caught my eye was exactly when it will end. The certificate that's presently being sent to any visiting web browser is displaying a "not valid after" date of October 26th of this year.

Now, we might expect Entrust to attempt to renew any certificates they can before the Halloween drop-dead date. And since the IRS's current certificate will need renewing before October 26th, it will be interesting to see whether they remain with Entrust, as they certainly could for another year, or whether a policy somewhere deep within the bureaucracy triggers a change. We'll see. And we won't have long to wait because by Halloween they will be recertified. We just don't know who will sign their certificate.

Jonathan said: "Hello, Steve. I found a connection from my iPhone to one of the polyfill-related domains, cdn.staticfile.org. There was one look-up in my NextDNS logs on June 24th. Obviously it would be difficult, if not impossible, to locate the source of the lookup on an iPhone. I looked for information on how to respond to this potential compromise, but all I can find is information for site operators, you know, remove dependencies on polyfill. I see no other connections to the known indications of compromise domains in my logs. What would you recommend at this point to make sure I'm not hacked? I'm thinking of wiping and reinstalling the OS, a backup, or starting fresh. Thank you." And he says: "From an undisclosed location near Washington, DC."

Leo: Oh ho, it was the President. Okay.

Steve: So it's 100% true that we don't know what we don't know. And the reason the Polyfill.io event was so significant was mostly how bad an attack could have been. But all indications are - again, within what we know - that for whatever reason, Funnull chose to only use this immense power they had to launch highly targeted and selective attacks against users of mobile devices who were selected by the make and model of the handset they were using and only when visiting specific websites.

Funnull's missed opportunity is the massively large bullet that we appeared to have dodged. Funnull may have imagined that their hack would never be discovered, so they may have been in no hurry to do more damage. And they likely figured that as long as they continued to deliver the proper polyfills to nearly everyone who asked, their deception would go unseen. So my point is, based upon everything we know, the actual likelihood that you, Jonathan, or I, or anyone would have ever been subjected to Funnull's malicious code truly seems vanishingly small. You know, I'm an avid iPhone and iPad user, and I haven't given it a second thought. None of the forensic analysis that's been done after this was discovered has revealed any more than those very tightly targeted attacks. It may have only ever been a handful of users who got this malicious Javascript.

But also Jonathan asked: "What would you recommend at this point to make sure I'm not hacked? I'm thinking of wiping and reinstalling the OS, a backup, or starting fresh." There's no indication that the malicious JavaScript, even if the targeting happened to match with you and somewhere you went, was exploiting a vulnerability in the platform you were using. So when we say it was malicious, we don't necessarily mean that it was exploiting a vulnerability. It's almost certain that even in those who were penetrated, nothing about their browser or OS was ever compromised. That's a whole different end. The attack would have just used JavaScript code running in the browser at that website to, whatever, grab their login credentials or their browser's session cookie to impersonate them, or something of similar value to the attackers.

As I noted last week, since the browser was loading the Polyfill.io code in the browser's first-party context and giving it access to the browser's DOM - you know, the Document Object Model, the webpage's guts - that code could do whatever it wished, but probably only within the bounds of what any JavaScript code could do. In other words, your browser and OS would not be damaged at all. Therefore, first of all, incredibly unlikely that you ever actually received any malicious JavaScript; and, even if you did, especially

on an iOS device, vanishingly small chance that a compromise was required. It just wasn't necessary in order to probably get what they wanted, which would have been login credentials or a session cookie, something like that.

Okay. From Bud in West Virginia. I want to share a longer-than-usual piece of thoughtful feedback from a listener of ours. By taking a fact-based look at Microsoft's actual past delivered behavior, he makes what I think is a factually supported case for Microsoft clearly placing their own profit well ahead of the needs of the users of their Windows desktop. And while, yes, okay, maybe that's obvious to all of us, the conclusion that he draws and what I think he predicts is worth looking at. So I'm going to share first what Bud wrote, and then I'll discuss it.

So he says: "Hi, Steve. I realize this is a bit long and tried the best I could for brevity. When I first heard about Recall, I thought it could be a useful tool, but also expected it to be a mess. So far I'd say that's accurate. And after listening to your coverage about Recall, I think it's going to be even worse than I originally thought.

"You've said multiple times recently that Microsoft has not shown malicious intent. But I believe that they have. Let's look at three Microsoft products, and then I'll share my thoughts about Recall and how it might be what finally makes me switch everything I'm responsible for away from Microsoft products and services. Yes, I believe it's that bad.

"First, let's look at Windows 10. I tend to be an early adopter," he writes, "and Windows 10 was no exception for me. When it was released, I was working in a small IT services company with customers in small business, local government, and home end users. As for Win10 upgrades, some people didn't want change, and some couldn't change due to a dependency on something not supported by Windows 10.

"Microsoft's rollout of Windows 10 basically went like this: First: 'Hey, Windows 10 is a free upgrade for 7 or 8.' Then: 'You haven't upgraded to 10. Let's schedule it.' And then: 'I'm going to schedule the upgrade unless you click in the fine print,' which he says tricked a bunch of users to upgrade. And finally, no notification, no choice, some users went to bed with Windows 7 or 8, and on their computer they woke up to have Windows 10."

He says: "Okay. Next let's look at OneDrive. Last year, Microsoft started asking users to back up their desktop and other folders to OneDrive. Then after saying no, some users found that when trying to delete something from their desktop, they'd get a message stating that items deleted from OneDrive could be recovered. Microsoft has now started asking in Windows 11 initial setup, but then turning 'ON' folder backup even if the user selects not to."

I'll just note that I've been listening to Paul Thurrott lamenting this more recent behavior of Microsoft's which has also been driving him crazy. Microsoft is, indeed, ignoring these settings even when they are arguably privacy oriented and should be entirely within the user's control. But as Paul keeps saying, they're just ignoring him.

So Bud continues: "And, finally, the kludge (wonderful word) that is Microsoft Edge. Microsoft Edge (Chromium) started off as a great browser." He says: "I used it for a few years. But now it's so bad that I'll use literally anything but Edge. There's too much content here to choose from so I'll just choose the latest that has impacted me."

He says: "I've worked in DevOps for several years and redeploy Windows VMs often. The startup screens for Edge have over the past couple of years gone from 'Please sign into your Microsoft account' to this infuriating mess." And he has four points. "Sign in to your Microsoft Account." Then, "Let's sign in to your Google account to pull in that data." Then, "We're scheduling to pull data from other browsers on a regular basis," which he

says is "enabled by default." And finally, "Let's make your experience better for you," and he says, "really meaning better for Microsoft to track you and target you with ads."

And he says: "Finally, the coup de grace. After already turning off the 'Let's make your experience better' setting, opening the browser sometime later will open a small notification that Microsoft has made your experience better anyway. And if you don't like it, go to Settings and change it, again." He says: "Every time I get that notification, I'm already typing, and some key hits OK. If you aren't paying attention, you could easily miss it.

"So how does this all come together and apply to Recall? Microsoft has clearly demonstrated that they can AND WILL pressure, trick, countermand, and/or silently change settings to what will benefit them. And Microsoft has heavily invested in AI and needs some return on that AI investment." He says: "I don't think they're just setting the stage for an ad-supported version of Windows. They are going to want all Windows systems to have Recall enabled so they can have hundreds of millions of computers that can be targeted for advertisings. And everything the user does, not only web browsing activity, will be monitored. It will be everything they do.

"Microsoft needs that AI return on investment, and this is how I think their deployment will likely go. First: 'Hey, Windows Recall is ready, and you should turn it on now.' Then: 'I'm going to go ahead and enable Recall. You can disable it in Settings.' And finally: 'Windows Recall now works on non-Copilot+ PCs! Let's enable it now!'" He said: "And when all that isn't enough, Microsoft will just silently enable it anyway and people won't know until search exclaims 'Your search is now enhanced by AI and Windows Recall!'"

He finishes with: "I sincerely hope Microsoft does not take this path. But given their track record as I've outlined it above, I think it's all too likely." He says: "I'm in the market to replace an old laptop and would love to get an EliteX-based system. But I'm waiting until Linux is an option for it and for AMD's next-gen system to be released because I simply do not trust Microsoft to be content with leaving Recall disabled. They have an established history of breaking the workarounds." And he says: "Looking at you, Edge!"

Then he says to me: "I'd like to hear your thoughts on this review of Recall and Microsoft's intentions given their history. Thanks again for all you and Leo and the other hosts do to provide great shows every week. Signed, Bud."

And I have to say it's difficult to argue with Bud's assessment. As I said at the start of this, he makes a strong evidence-based case for what Microsoft seems very likely to do with Recall in the future. One thing we do know is that it's been very clear from everything they've said that they are very determined to push Recall onto the desktop. Which really does beg the question, why? What's in it for them? Why is Microsoft so anxious to push everyone into using Recall, if it's just to give us better search? That doesn't really track. I've heard from other listeners whose opinions more align with Bud's in this regard, so I wanted to share Bud's well-reasoned perspective. And I will reiterate that should or when this comes to pass, I will make a definitive Recall blocker available as a piece of lightweight GRC assembly language freeware.

And one last note. Thomas Tomchak said: "I have a disproportionate amount of joy for you having a newsletter via email. Thank you for putting the work in to make it happen and to do so on your terms."

Leo: Aw.

Steve: And so Thomas, I just should say I just chose one quote. I'm getting constant approbation from our listeners who are just delighted to receive GRC's weekly emailing in advance of this podcast. I did have, for what it's worth, a problem this week which was interesting. The emailing contains a thumbnail of the Picture of the Week, which as we know was that crazy bathroom door lock issue. About 80 or so of our recipients had the email bounce, claiming that it contained a virus. Apparently ClamAV had a false-positive match on the binary of the jpeg and thought it was malicious. And so 80 of our listeners, although that's 80 out of currently 6,471 who have signed up for the...

Leo: Conveniently, almost exactly one-eighth.

Steve: Yup. So about - anyway, I just wanted to say, for those of you who did not receive it, though you are signed up, it's because your email provider is using an antivirus system, ClamAV, which identified the thumbnail as being a virus. So it bounced. I had, like, two last week, and 80, something like 80 this week. So, and I'm using the same template. So, like, nothing changed except a different picture. And the only reason I can see that AV would trigger would be when you look at the body of the email, it is a binary blob because that's the jpeg thumbnail. And, you know, I could omit it, but it's fun to have that picture in email. And other people get to have pictures in email, so...

Leo: Why not me?

Steve: I'm just going to chalk it up to hopefully a rare occurrence.

Leo: One 80th, of course, not one eighth.

Steve: I think you'll get email.

Leo: Yeah, one 80th of your entire group.

Steve: Right.

Leo: Let us break, and then the meat of the matter, the Snowflake's Chance. We'll talk about the Snowflake breach. Who's at fault for that after all, the AT&T breach? Okay, Steve. Let's talk about Snowflake.

Steve: Okay. So there's undeniable logic in the proposition that a third-party organization specializing in some aspect of business operations can, within a limited sphere, do a better job and a more cost-effective job than a company whose business is not doing that. So the idea of farming out to a subcontractor some chunk of work becomes appealing, when that's not your business's main focus.

Like, for example, when a building is being built, you use a subcontractor who specializes in laying foundations to do that work. You don't ask your painter to do that. And the commercial plumbers install the plumbing, and the HVAC guys run the air ducting and install the equipment on the roof, and so on. So from a theoretical standpoint the model is sound. It can and has gone wrong, of course. If a contractor is discovered to be doing

substandard work, it's certainly prudent to go back and look at the previous buildings they worked on to determine whether those might also have been impacted.

Now, as we know today, the "Cloud" is all the rage. I've told the story of participating in a DigiCert customer summit seven years ago where all of the other techies looked at me like I had two heads when I casually mentioned my rack of servers at Level 3, one of them saying to me, "Steve, no one does hardware anymore." No one does hardware anymore. Right.

What's been happening for at least the past several decades or more is that a few nerds who know each other will get together over some pizza to discuss ways to make a bazillion dollars. The framework of their idea is nothing new: Create a business plan and present it to some venture capitalists in order to obtain seed capital and form a classic start-up. Work 24/7 to create something everyone needs, then start it running. Watch it grow, create demand, then either take it public or sell it off to a much bigger fish. The venture capitalists are happy, the cofounders are rich, and everyone wins.

So in a world where I'm told that "no one does hardware anymore," it was only natural for those nerds to turn their attention to offering various sorts of cloud services. And the model there is more intoxicating than anywhere else, since not only do their future customers not want to "do hardware," neither do they. And they don't need to since massive data centers already exist where "doing hardware" is all they do - again, another example of increasing specialization.

So these nerds write a bunch of code to do whatever it is they think companies will not be able to live without once they see what their new service is capable of doing for them. They rent some servers, spin up a bunch of virtual machines, launch their website, make an offer for trying it for free before committing, and start looking for and signing up new customers.

Okay, now, I wrote everything I've just shared before I went to Wikipedia to see what Wikipedia had specifically to say about Snowflake. I promise that I really did write all of that with zero specific knowledge of Snowflake. So here's what the start of Wikipedia's page on Snowflake says. Wikipedia writes: "Snowflake Inc. is an American cloud computing-based data cloud company based in Bozeman, Montana. It was founded in July of 2012 and was publicly launched in October 2014 after two years in stealth mode. The firm offers a cloud-based data storage and analytics service, generally termed 'data-as-a-service.' It allows corporate users to store and analyze data using cloud-based hardware and software.

"The Snowflake service's main features are separation of storage and compute, on-the-fly scalable compute, data sharing, data cloning, and support for third-party tools. It has run on Amazon Web Services since 2014, on Microsoft Azure since 2018, and on the Google Cloud Platform since 2019. The company was ranked first on the Forbes Cloud 100 in 2019. The company's initial public offering raised $3.4 billion in September of 2020, one of the largest software IPOs in history," writes Wikipedia.

"Snowflake Inc. was founded in July 2012 in San Mateo, California by three data warehousing experts, two who previously worked as data architects at Oracle Corporation, and the third a cofounder of a Dutch start-up VectorWise. The company's first CEO was Mike Speiser, a venture capitalist at Sutter Hill Ventures." So pretty much exactly what I said is the way this all happens these days.

So the point I can now make from what was my first blind writing, without any, you know, specific knowledge of Snowflake, is that, indeed, this is the way today's cloud-based service ventures are being born. And, as Wikipedia's details have shown us in this case, the founding three were absolutely correct about the need for and the appeal for

their service. Since we're going to be talking about what happened in a minute, it's worth getting a little more specific information about this company.

So Wikipedia continues from where I had left off: "In June 2014, the company appointed former Microsoft executive Bob Muglia as CEO. In October 2014, it raised $26 million and came out of stealth mode, being used by 80 organizations. In June of 2015, the company raised an additional $45 million and launched its first product, its cloud data warehouse, to the public. It raised another $100 million in April 2017. In January 2018, the company announced a $263 million financing round at a $1.5 billion valuation, making it a unicorn." For those who don't know, a unicorn is a startup company valued at over $1 billion which is still privately owned and not listed on any market. Wikipedia says: "In October 2018, it raised another $450 million in a round led by Sequoia Capital, raising its valuation to $3.5 billion.

"In May of 2019, Frank Slootman, the retired former CEO of ServiceNow, joined Snowflake as its CEO; and Michael Scarpelli, the former CFO of ServiceNow, joined the company as CFO. In June 2019, the company launched Snowflake Data Exchange. In September 2019, it was ranked first on LinkedIn's 2019 U.S. list of Top Startups. On February 7th, 2020, the company raised another $479 million. At that time, it had 3,400 active customers." Okay? Four and a half years ago, 3,400 active customers. "On September 16th, 2020, Snowflake became a public company via an initial public offering, raising $3.4 billion, one of the largest software IPOs and the largest to double on its first day of trading."

So four and a half years ago, back in February 2020, Snowflake had 3,400 active customers, and the sky's the limit. Everything looks great. We can presume that four and a half years later that number has only grown. So I wanted to start by painting that generic picture of the relatively new phenomenon of an entirely cloud-based industry, of which Snowflake is a perfect example, because the events and finger-pointing in the aftermath of Snowflake's apparent inability to protect many of its customers' vast troves of data, much of it sensitive, suggests that we're not yet fully equipped to deal with the consequences of this new and essentially "virtual" cloud-based industry. A ton of information about what can now only be described as an historic data breach exists on the Internet. So I've spent a great deal of time following the links and reading original sources in an attempt to make sense of what happened.

I think I finally have it worked out, and it's not quite the narrative that has taken hold throughout the industry due to a bit of subtlety as well as contracts and non-disclosure agreements. Snowflake is blaming its customers for having their Snowflake login credentials used to log into their Snowflake accounts, noting that it was only those customers who did not have their logins protected by multifactor authentication that had been breached. In other words, it appears that Snowflake is blaming their own customers for having weak authentication security - while, I'll note, Snowflake did not require any stronger login authentication, as it certainly could have.

But it seems to me that the real question, which Snowflake appears to want to avoid answering by deflecting about multifactor authentication, while its security contractors may be bound by agreements not to disclose, is how were many hundreds of its customers' login credentials obtained by these attackers in the first place? The facts strongly suggest that something happened where, in short order, attackers obtained the login names and passwords belonging to a large number - hundreds - of Snowflake's customers. Where present, the attackers were apparently unable to obtain the accounts' MFA secrets, which is why MFA protected those customers who were using it. But somewhere around 350 of Snowflake's customers who were not using MFA suddenly found that all of the proprietary data they had shared with Snowflake had been exfiltrated to parts unknown.

So whose fault was it? Was it Snowflake's customers, for not extra-protecting themselves from what appears to be a major precipitating breach of authentication credentials at Snowflake? Or did Snowflake make some mistake themselves - which, to be clear, they are denying strongly - and that that preceding breach allowed a large set of their customers' login credentials to fall into the hands of the bad guys?

We know mistakes happen. That's a fact. But the narrative that's taken hold in the industry, which many articles quote Snowflake's spokesperson's saying, is that the actual fault lies with Ticketmaster, with Advance Auto Parts, with Santander Bank, with LendingTree, and now with AT&T, as well as apparently more than 340 others, for not using multifactor authentication. That's a nice sleight of hand on Snowflake's part, but I'm not sure it's fair.

Security researcher Kevin Beaumont often summarizes things with more technical detail than other publications. In this case, back toward the beginning of June, under his headline "Snowflake at centre of world's largest data breach," Kevin posted on Medium: "Cloud AI Data platform Snowflake are having a bad month, due to teenage threat actors and cybersecurity of its own customers, and its own cybersecurity, too, in terms of optics. There are several large data breaches playing out in the media currently. For example," he writes, "Ticketmaster owner Live Nation filed an 8-K with the SEC for potentially the largest data breach ever, claimed to be 560 million customers. They finger Snowflake as part of the data breach." Kevin cites TechCrunch's article with the headline: "Live Nation confirms Ticketmaster was hacked, says personal information stolen in data breach."

Then Kevin says: "Additionally, incidents are running at multiple other cyber companies who are Snowflake customers where full databases have been taken." He says: "I've spoken to people in multiple industries at large corporations where they've had significant data exfiltration in May via Snowflake. The Australian security services have issued an advisory: 'High Alert / Act Quickly!' They say they are 'aware of successful compromises of several companies using Snowflake environments.'" He says: "Snowflake themselves have put out Indicators of Compromise for 'threat activity' over the weekend, saying to look for connections into their platform from the user agent 'rapeflake.' Additionally, a threat actor claims they gained access to Snowflake itself and their customers using infostealers."

Okay. So let's pause here because what happened has been interesting. The security research firm Hudson Rock first told the story of the penetration of Snowflake, but quickly received a takedown order from Snowflake's legal beagles. You know, 'We're going to sue you if you don't stop saying this.' So Hudson Rock complied, and the industry was then forced to reference the Internet Archive's Wayback Machine record of their write-up, until it was hit with a similar order, requiring it to block that URL from access. So not really a good look for Snowflake. What Hudson Rock had to say was interesting, so we'll circle back to that in a minute.

Referring to what Kevin read in Hudson Rock's piece, he wrote:

"The threat actor makes various claims which sound questionable; but, well, Snowflake have confirmed some of it is true while crowing to the media and customers that the blog is not true. It is Schrdinger's Blog. The threat actors here, from what I've managed to establish, were a teen crimeware group who've been publicly active on Telegram for a while."

Leo: Wow.

Steve: You know, thus rapeflake or whatever it was that they called this, their agent. Was it rapeflake? Yeah, rapeflake. As, you know, snowflake, rapeflake. So okay. In other words, you know, a bunch of kids did all this damage. When Kevin writes "Let's Recap," he says: "We have what appears to be the world's biggest data breach in terms of impacted individuals playing out with Snowflake as the vendor linking the victims. A lot of data has gone walkies. Snowflake, for those won't know, is an AI data platform where you shove vast amounts of data in and then use it. It allows you to do this with effectively no security."

He says: "I feel bad for Snowflake on a human level as they're in a bad situation. This is a potentially business-ending event for them. So they have to use every lever possible to point the fingers at their own customers as being negligent over 'rapeflake' activity to avoid responsibility. And to be clear, some of this is their customers' responsibility. But also, Snowflake have to own this issue and face straight into it to survive, as there's an extremely high chance this is going to play out publicly over coming weeks and months."

And boy, was Kevin prescient about that one. He wrote this more than a month before the AT&T breach announcement. Then he writes - and this is so perfect. He says: "Note that in the age of SaaS (Software as a Service), your providers will throw you under the bus to save themselves. When you transfer your security risk to a provider, they don't accept your risk. They just take your money." He says: "What you're sold versus what you get often don't align." He says: "I've worked for a cloud provider. You don't want to see how the sausage is made. And there's no real accountability for the provider. There will be much more of this to come with cloud data providers in the future, is what I'm saying.

"So what actually happened? Despite Snowflake saying the Hudson Rock blog is inaccurate," and he says, "and parts most probably are, the Snowflake credentials bit is accurate. Snowflake say: 'We did find evidence that a threat actor obtained personal credentials to and accessed demo accounts belonging to a former Snowflake employee. It did not contain sensitive data. Demo accounts are not connected to Snowflake's production or corporate systems. The access was possible because the demo account was not behind Okta or Multi-Factor Authentication (MFA), unlike Snowflake's corporate and production systems.'"

And I'm just reminded, didn't LastPass go on and on about how safe everything was because their development systems were completely isolated from their production and corporate systems? That sounds like a familiar tune we've heard before. Whoops.

He says: "Snowflake have incident response stood up, with CrowdStrike and Mandiant involved. They say the cause of the malicious activity - in other words, database downloads - is 'This appears to be a targeted campaign directed at users with single-factor authentication. As part of this campaign, threat actors have leveraged credentials previously purchased or obtained through infostealing malware.'"

Okay. So to me this is curious, since they're not saying from where a huge number of their customers' single-factor authentication credentials may have been "infostolen." There's only one place in the entire world where all of those otherwise completely independent customer credentials would all be gathered into one place. I wonder where that could be? As Kevin wrote: "In the age of SaaS, your providers will throw you under the bus to save themselves."

Okay. So as he says: "So what happens, essentially, is infostealers were used to gain access to Snowflake databases using their customers' stolen credentials, using the client name rapeflake." And then he said: "Side note to threat actor over that name. Really?" Anyway, he finishes: "Snowflake themselves fell into this trap by both not using multifactor authentication on their demo environment and failing to disable an ex-

employee's access. Stuff happens. Incidents happen. And while Snowflake may present themselves as having no platform breach, they themselves also fell into the same problem, and in terms of optics it isn't great as they can point out customers messed up, but then they messed up, too."

So he then went on a little bit of a digression about what he feels is a hugely important topic of "infostealers." And since it's Kevin, I'm going to share that. He said: "You may know about infostealers as I recently wrote about them being a huge threat when it comes to Microsoft Copilot+ Recall allowing full data threat of everything you've ever viewed, a feature you should absolutely disable in Windows 11."

Then he says: "Mandiant themselves have this to say about infostealers this weekend." And then I grabbed a picture of Kevin's snap from Mandiant's site, where he says: "Here are some of Mandiant's observations related to infostealers from the past few years. Since the beginning of 2020, employees and contractors working from home increasingly use their personal computers to access corporate systems. People often synchronize their web browsers on their work computers and personal computers. People, or their children, sometimes inadvertently install software laced with infostealing malware on their personal computers. The malware can capture credentials from their web browsers. Threat actors opportunistically search for corporate credentials stolen by infostealing malware to use them to compromise enterprises, steal data, and conduct extortion."

In other words, although it's not a straight line, it's a series of interconnections. A worker at home is using his personal machine on the enterprise network. The personal machine is synchronizing browsers with the enterprise network at the other end. Someone may install infostealing malware on the home machine. That home machine can steal the local credentials from the browser which, being synchronized with the enterprise browser, gets the enterprise credentials. And then the infostealing machine on the home computer is able to whisk those off somewhere.

So Kevin says: "If you use Snowflake, you need to first of all enable multifactor authentication and tighten authentication to your database as a top priority. Then you need to go back and look at the access logs on Snowflake itself and check who's been using your data." He says: "You cannot rely on Snowflake doing this for you." He says: "Infostealers are a significant problem. It has long since outpaced botnets and so forth in the real world. And the only real solution is robust multifactor authentication, and ideally getting rid of passwords altogether by replacing them with secure authentication." In other words, Passkeys.

He says: "There are companies offering services where you can buy your own stolen credentials back" - whoa - "and then you can change users' passwords." He says: "I don't like this approach. The reason is those vendors often buy those credentials from 'credential brokers,' which translates to funding the criminal hackers who steal them in the first place. As a customer you end up proxy funding the threat actors you're trying to deal with. Additionally, it is a huge user impact to have their password changed, and it doesn't fix the problem."

He says: "Tightening authentication fixes the problem. Ask the Snowflake victims how they have fixed the problem. It's through robust multifactor authentication. The wider problem is that something is wrong at Snowflake when it comes to authentication. Snowflake themselves fell victim to this incident, albeit with a demo tenant. They need to, at an engineering and secure-by-design level, go back and review how authentication works, as it's pretty transparent that, given the number of victims and the scale of the breach, that the status quo has not worked. Secure authentication should not be optional. And they've got to be completely transparent about steps they're taking off the back of this incident to strengthen things.

"For cloud providers in general, they need to be more robust in terms of secure defaults or risk being dragged into this kind of situation. For Microsoft" - I love it that he just couldn't resist finishing with this. "For Microsoft, they need to recall Recall, or they will pour petrol onto the flames and make the infostealer problem far worse."

After Kevin posted this piece, he added a quick follow-up. He said: "People are pinging me to say there's more to this story than I've disclosed." And he says: "I know. It will be a developing story, and all eyes are on Snowflake." So maybe, for example, there was some knowledge of the AT&T problem. So anyway, you know, Snowflake insists that it didn't happen that they were breached. Maybe that's true. Seems suspicious to me since all of a sudden someone got a hold of 350-plus of their customers' non-MFA single-factor authentication login data and used it to breach their technologies. It'll be interesting to see if anything more happens and, you know, how Snowflake fares. They are certainly a huge cloud provider.

It's also interesting, and here a little egg is on AT&T, the data that was stolen from AT&T was apparently two years old. That is, this was back from 2022. AT&T left old, 110 million customers' worth of individual transaction data at Snowflake from two years back and, you know, could have pressed a button to delete it, but didn't. So, you know, plenty of blame to go around. But, wow, we now have broken all records in terms of amount of data lost in a massive breach.

Leo: So do you think that the credentials of those 350 companies were revealed, or that there is something - more likely there's a flaw in the authentication process; right? That whatever it is takes in the password and then lets the person in wasn't working properly? There was a hole in it? We don't know, do we.

Steve: I don't think the - everybody who is surrounding this thinks something is fishy.

Leo: Yeah.

Steve: What Snowflake - it is hard to keep saying that name. What a name. Maybe they'll change it.

Leo: They thought it was funny at the time, you know. Hey, we're a unicorn, Snowflake, ha ha.

Steve: What they're saying is that their customers were the victims of infostealers which found Snowflake credentials on their customers' computers.

Leo: Okay.

Steve: And then, because those customers weren't using multifactor authentication, just finding static authentication, username and password...

Leo: Yeah, was enough.

Steve: ...allowed them to log into Snowflake.

Leo: Okay.

Steve: If that's the case, then why not everything else? I mean, and why all of a sudden, you know, 340, 350 customers? It seems much more likely that Snowflake was infiltrated, and the database of those, you know, that authentication was exfiltrated and then used because those customers did not have multifactor authentication...

Leo: That was all they needed.

Steve: It was used to log in as them and grab all their data. We just don't know.

Leo: We don't know. It's possible that the hacking group was actually looking for Snowflake credentials on all those customer computers.

Steve: Yes. It absolutely is possible.

Leo: That they have some sort of, you know, they knew they were targeting Snowflake, and they were looking for that kind of credentials. That's why it was all Snowflake.

Steve: Yes.

Leo: I mean, it's unknown. Somebody needs to come forward and say what happened. But Snowflake probably...

Steve: Well, actually, The Record, the publication The Record, said this. They said: "According to the original post, the intruders were able to sign into a Snowflake employee's ServiceNow account using stolen credentials, and from there were able to generate session tokens."

Leo: Ah.

Steve: Hudson Rock wrote: "To put it bluntly, a single credential resulted in the exfiltration of potentially hundreds of companies that stored their data using Snowflake, with the threat actor himself suggesting" - the threat actor himself suggesting - "400 companies were impacted. In a post on Friday, Snowflake did not respond directly to the researchers' claims, but denied that a vulnerability within its systems was to blame for the accessing of customer data. The company said it is 'investigating an increase in cyber threat activity'."

Leo: A large increase.

Steve: Uh-huh. So basically Hudson Rock posted this claim by the threat actor themselves, and they received a legal takedown notice.

Leo: Oh.

Steve: And then the web archive was similarly forced to block their archive link.

Leo: Well, that smacks of cover-up.

Steve: Yes.

Leo: I mean, I guess you could say for security reasons we don't want anybody to know how they did this.

Steve: Or PR reasons.

Leo: Or PR reasons. Wow. Oh, I hope we get to the bottom of this at some point. I'm sure, if we do, you will let us know.

Steve: Absolutely.

Leo: What a great story. Great in an interesting way, not a good way for anybody involved. Thank you, Steve Gibson. Once again, elucidating the dark corners of the Internet.

Steve: Bye.