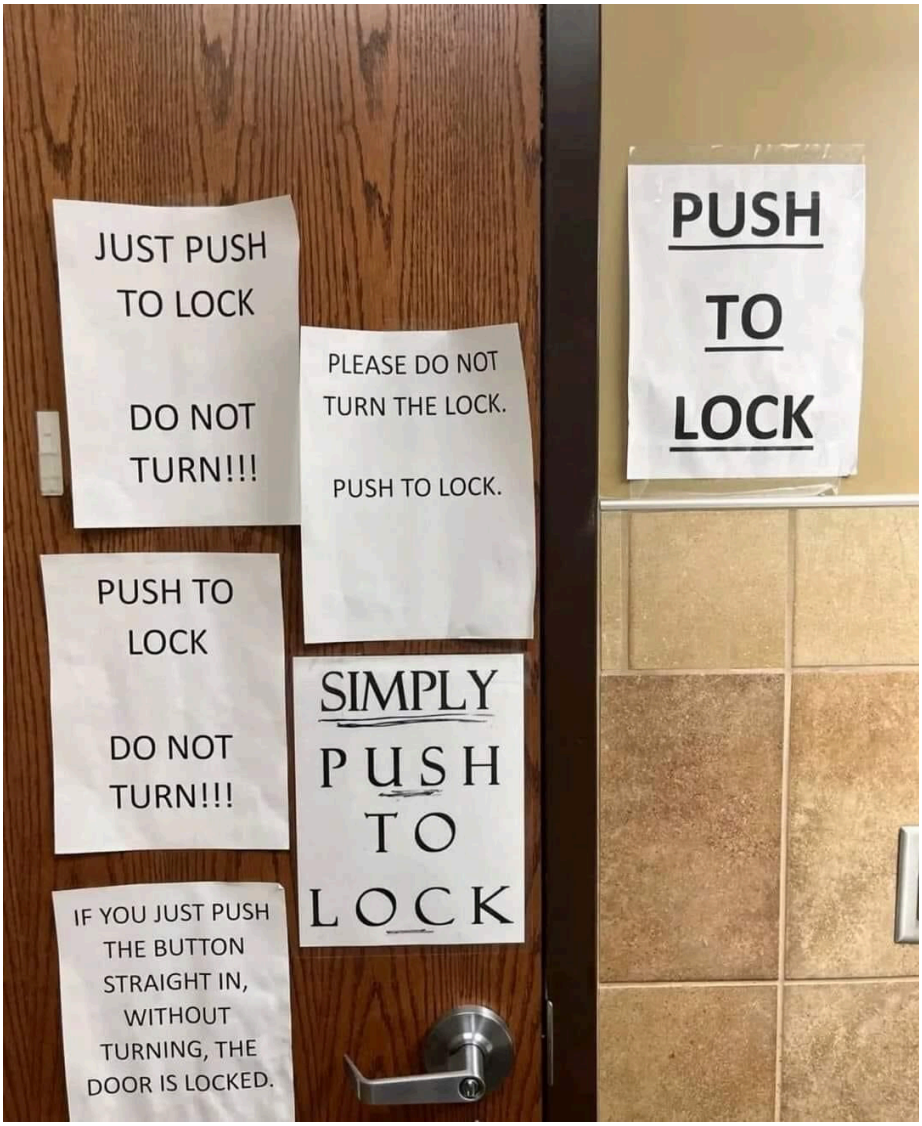
Security Now! #983 - 07-16-24

A Snowflake’s Chance

This week on Security Now!

- 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?

Does anyone wonder how to lock this bathroom door?



Security News

Using Content Delivery Networks Safely

Looking back upon last week's "Pollyfill.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 3rd parties such as high performance content delivery networks, CDNs. It was not my intent to rain on the idea of CDNs in general for this purpose, because the web's designers have made ample provisions for safely pulling code into webpages 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 3rd-party content **can** be delivered safely and why the polyfill.io facility was never able to take advantage of it.

The formal name for the facility is "Sub-Resource Integrity" abbreviated SRI where the concept and implementation could hardly be clearer, cleaner and simpler – as the best things are: The same HTML <script> tag that contains the URL of some remote 3rd-party code or stylesheet that the browser is being asked to load, can also, optionally, contain another name/value pair:

integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKpYfRvH+8abtTE1Pi6jizo"

The format is the word "integrity" followed by an equals (=) sign then any one of the strings "sha256", "sha384" or "sha512". That hash specifier is followed by a dash, followed by the specified hash of the expected URL resource hash encoded from binary into base64 ASCII.

So here's what all that means: When a web designer wishes to pull some remote resource from a content delivery network, or anywhere, really, that they do not directly control, 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: https://www.srihash.org/ or they can use 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 and return the snippet of script tag code that contains the proper matching hash.

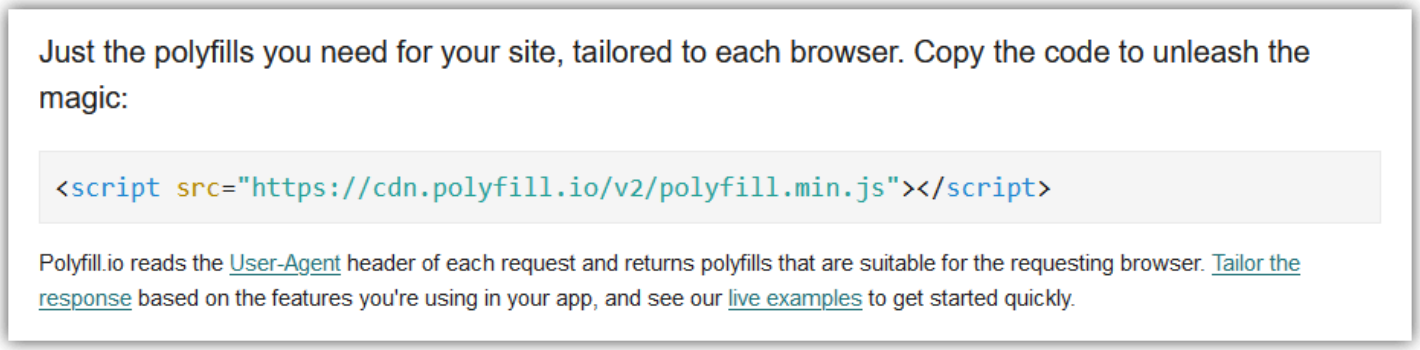
So our designer drops the URL of the jQuery library they wish to use into srihash.org and receives the hash. They then add this hash, along with the "integrity" keyword into the jQuery fetching script tag in their webpages they are henceforth protected from any modification of that code.

When their page is delivered to a user's browser, the browser reads the pages HTML code, 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 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 this does is very nicely and cleanly allow web designers to protect their site's users from both inadvertent or deliberate alteration of the resource they're requesting. Also note that since code libraries are constantly evolving – jQuery is currently at v3.7.1 – it's also necessary for the jQuery or whatever library to specify the exact version that is 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 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 invoking tag.

Okay. So now we 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 for super-fast content delivery without delay can be used without risk.

But as I said, earlier, unfortunately this very slick protection was not available to users of polyfill.io. I touched on this last week but since it wasn't our focus I didn't elaborate or highlight its significance. So just now I used our trusty Web Archive's Wayback Machine to show the polyfill.io website homepage. They were clearly very proud of what they'd created. The home page of the site says:



Then they provide a sample bit of HTML with a <script> tag requesting the polyfill resource. Underneath the code sample they explain further: Polyfill.io reads the User-Agent header of each request and return polyfills that are suitable for the requesting browser. Tailor the response based on feature you're using in your app and see our live examples to get started quickly.

In other words, there isn't one polyfill.io code library. And there are no version numbers. 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 website visitor was using. This always made it actively hostile to the web's subresource integrity system, which prevented any of polyfill.io's great many users from supplying a hash of the code they would receive... since what polyfill.io returned would be entirely unpredictable.

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 3rd party, should add to their bag of tricks.

<<sponsor insert>>

The CDK Global Ransomware Attack

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:

> *Hello Mr. Gibson,*
>
> *My name is Shawn and I am an automotive technician at a GM dealership and have been listening since about 2016. I love when your world of security crosses over to my world of automotive. My dealership (as well as thousands of others) is affected by the CDK cyber attack 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 (we get paid by what we do and this is slowing everything down as we have to go back to manual RO's and quotes, lowering my booked hours). Thanks, Shawn.*

And also on June 21st I received the following from:

> *Steve thanks for all the years of podcasts. I have 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 are 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. The delivery box truck that stopped was almost empty today. (I only got my parts today as they had already been ordered and invoiced prior to the issue).*
> *Thanks*
> *Alan Alberg, Alberg Auto*

And 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 stand-still by this cyber attack. 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 the of saying, "There is no cloud, you are just someone else's computer." Usually that other computer is better secured than what 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, NC*

So what's the scoop on this? We have a situation were 15,000 operating dealerships were dependent upon a single MSP, a managed service provider or SaaS, software as a service for all of their paperwork processing. We'll be talking more about software as a service when we get to 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 clean up crew when several types of problems arise. There are still systems and procedure hiccups that happen today, but thanks to technology and automation they are fewer in number. Then came the CDK cyber attack.

This CDK cyber attack 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. 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.

Why did the CDK cyber attack happen?

There was once a company called ADP Dealer Services who were a great DMS provider. (DMS is Dealer Management System.) 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 [wait for it, yep...] *private equity investments.*

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 (hackers) are the first sent off to exile.

When there is 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 a few things on this list have to be true":

- *They have no backups, or*
- *If they do have backups, they are outdated or never tested, which is effectively the same as having no backups.*
- *No one knows how to restore backups.*
- *There is no disaster recovery plan, or if it exists it is outdated to the point of uselessness.*
- *Multiple single points of failure are baked into the infrastructure.*
- *They have no idea how compromised they are.*

I am very angry about how ADP Dealer Services, a once great company, has been raped and pillaged by private equity.

The real pain is suffered by the rank and file at 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.*

Huh. Information Security doesn't seem like such a waste of money now, does it?

How did the CDK cyber attack happen?

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 much 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.

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. Kathi continues...

Theoretically, a mature Dealer Management System provider should be able to lose any single critical part of your core business and be able to restore functionality within 24 hours (barring a massive natural disaster/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 and even months to get all the data in order after they receive the keys to the ransomware. The database will likely have holes in it that will add to the arduous restoration process.

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. When operations is functional again, all the business they produced — new and used car sales, service, parts, internals, warranty — anything that happened during the down time, 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+ 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"), then all entries should land in their respective GL accounts. Schedules and other GL 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 the 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, it will ultimately come together.

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?"

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:

- *Dealers for impeding commerce and negligence in data loss (among other things).*
- *Consumers for the massive data breach of extremely sensitive information.*
- *Employees for data privacy and lost compensation.*

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.

In subsequent reporting, CNN Business reported, under their headline "How did the auto dealer outage end? CDK almost certainly paid a $25 million ransom" they wrote:

CDK Global, a software firm serving car dealerships across the US 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 21, 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.

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.

Will the payment of that $25 million affect CDK's behavior going forward? Who knows. The greatest cost is likely 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 – 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 private equity owners having 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. Mistakes happen. Could profiteering ownership have been a contributing factor? 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 and without paying a multi-million dollar ransom was a good one. But we have no idea why that was the case.

And to my mind that's not really 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.

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: "Just press this button and follow the on- screen 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 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 now that the network and the automation that it provides is back up and business can resume as normal. CDK's CEO has apologized, has 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.

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. 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 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 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.

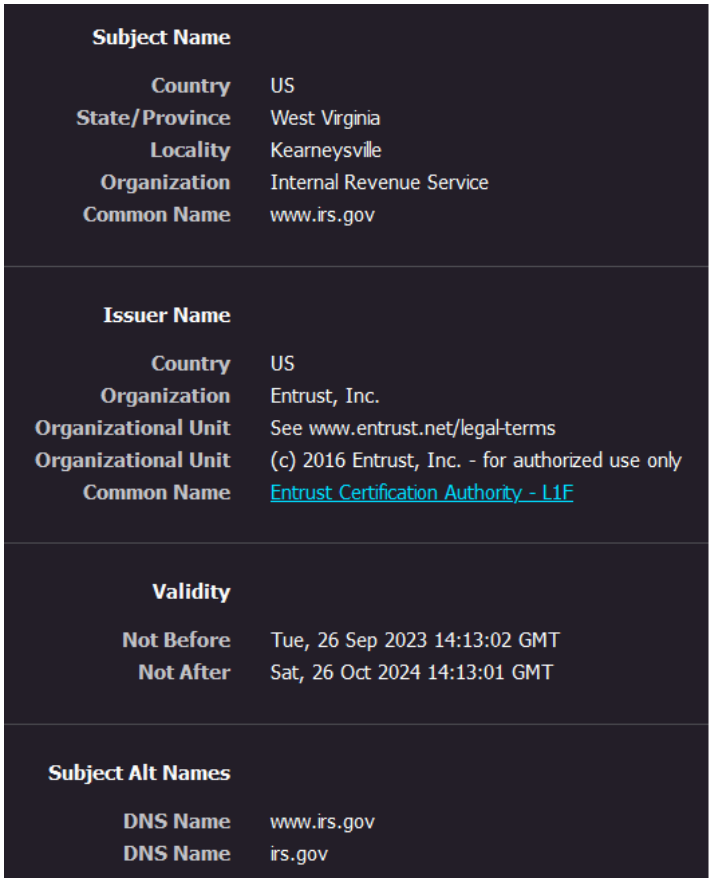
Thanks to the network effects of this event, a great deal of press coverage and attention was given to this. Let's hope that some lessons were learned to better prepare other similar organizations to respond if it should happen to them.

<<sponsor insert>>

Closing The Loop

Knox North / @newtronic

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 http://irs.gov and guess who issued their cert?



Subject Name	
Country	US
State/Province	West Virginia
Locality	Kearneysville
Organization	Internal Revenue Service
Common Name	www.irs.gov

Issuer Name	
Country	US
Organization	Entrust, Inc.
Organizational Unit	See www.entrust.net/legal-terms
Organizational Unit	(c) 2016 Entrust, Inc. - for authorized use only
Common Name	Entrust Certification Authority - L1F

Validity	
Not Before	Tue, 26 Sep 2023 14:13:02 GMT
Not After	Sat, 26 Oct 2024 14:13:01 GMT

Subject Alt Names	
DNS Name	www.irs.gov
DNS Name	irs.gov

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.

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.

Jonathan

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 (remove dependencies on polyfill...). I see no other connections to the known IOC 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, Undisclosed location near Washington DC.

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 (or anyone) would have ever been subjected to FunNull's malicious code seems vanishingly small. 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 tightly attacks.

Also, Jonathan, you 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 any vulnerability in the platform you were using. So it's almost certain that even in those who were penetrated, nothing about their browser or OS was compromised. The attack would have just used JavaScript code running in the browser at that website to grab their login credentials or browser session cookie, 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-part context and giving it access to the browser's DOM – the webpage's guts – that code could do whatever it wished, but probably only within the bounds of what any JavaScrip code could do. In other words, your browser and OS would not be damaged at all.

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 is, I think, a factually supported case for Microsoft placing their own profit well ahead of the needs of the users of their Windows desktop. So I'll share what Bud wrote, then let's discuss it a bit:

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 have said multiple times recently that Microsoft has not shown malicious intent. But I believe that they have. Let's look at 3 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. Yeah, I believe it's that bad.

First let's look at Windows 10. I tend to be an early adopter and Win10 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:

1. *Hey, Windows 10 is a free upgrade for 7 or 8.*
2. *Then, "You haven't updated to 10. Let's schedule it."*
3. *And then "I'm going to schedule the upgrade unless you click in the fine print" which tricked a bunch of users to upgrade.*
4. *Finally... No notification, no choice, some users went to bed with Windows 7 or 8 on their computer and woke up to have Windows 10.*

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 the 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 his control.

Bud continues:

And finally, the kludge (wonderful word) that is Microsoft Edge. Microsoft Edge (Chromium) started off as a great browser. 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.

I've worked in DevOps for several years and redeploy Windows VMs often. The startup screens for Edge have over the past couple years gone from "Please sign in to your Microsoft account" to this infuriating mess:

1. *Sign in to your Microsoft Account*
2. *Let's sign in to your Google account to pull in that data.*
3. *We're scheduling to pull data from other browsers on a regular basis (enabled by default)*
4. *Let's make your experience better for you...really meaning better for Microsoft to track you and target you with ads.*

And 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. 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 their AI investment.

*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.*

Microsoft needs that AI ROI and this is how I think their deployment could likely go:

1. *Hey, Windows Recall is ready and you should turn it on now!*
2. *I'm going to go ahead and enable Recall... You can disable it in settings*
3. *Windows Recall now works on non-CoPilot+ PCs! Let's enable it now!*

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!"

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. 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 or for AMDs 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 work-arounds — looking at you, Edge!

I'd like to hear your thoughts on this view of Recall and Microsoft's intentions given their history. Thanks again for all that you, Leo, and the other hosts do to provide great shows every week. – Bud

It's difficult to argue with Bud. As I said at the start of this, he makes a strong evidence-based case for what Microsoft **may** attempt to do with Recall in the future. One thing we do know is that it's been very clear from everything they have said that are very determined to push Recall onto the desktop... which really does beg the question, why? Why is Microsoft so anxious to push everyone into using Recall?

I've heard from other listeners whose options are 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.

Thomas Tomchak

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.

<<final sponsor insert>>

A Snowflake's Chance

There's undeniable logic in the proposition that a third party organization, specializing in some aspect of business operations can, within that limited sphere, do a better and more cost effective job than a company whose business is not doing that. So the idea of farming that out to a subcontractor becomes appealing. When a building is being built, you use a subcontractor who specializes in laying foundations to do that work. You don't ask a 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, however. 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 be impacted.

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 said: "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 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 some new thing that 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 co-founder 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 corporations 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 customers.

I wrote everything I've just shared **before** I went over to Wikipedia to see what Wikipedia had to say about Snowflake. I promise that I really did write all of that with **zero** specific knowledge of Snowflake. So here's the start of Wikipedia's page on Snowflake. They say:

> *Snowflake Inc. is an American cloud computing–based data cloud company based in Bozeman, Montana. It was founded in July 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 2020, one of the largest software IPOs in history.

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 co-founder of a Dutch start-up Vectorwise. The company's first CEO was Mike Speiser, a venture capitalist at Sutter Hill Ventures.

The point I can now make from my first blind writing, without any 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 of their service. Since we're going to be talking about "what happened" it's worth getting a little more specific information about this company. Wikipedia writes:

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 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. [A unicorn is a startup company valued at over US$1 billion which is privately owned and not listed on a share market.] In October 2018, it raised another $450 million in a round led by Sequoia Capital, raising its valuation to $3.5 billion.

In May 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 7, 2020, the company raised another $479 million. At that time, it had 3,400 active customers. On September 16, 2020, Snowflake became a public company via an initial public offering (IPO) 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 3400 active customers. We can presume that four and a half years later that number has only grown. 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 sensitive data 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 links and reading original sources in an attempt to make sense of what happened.

I think I finally have it worked it 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 login to their Snowflake accounts, noting that it was only those customers who did **not** have their logins protected by multi-factor 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** 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 multi-factor 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 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** also 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 – which, to be very clear, they are denying – that allowed a large set of their customers' login credentials to fall into the hands of the bad guys?

Mistakes happen. We know that's a fact. But the narrative that's taken hold in the industry, which many articles quote Snowflake's spokesperson saying, is that the actual fault lies with Ticketmaster, Advance Auto Parts, Santander Bank, LendingTree, and now AT&T, as well as apparently more than 340 others for **not** using multi-factor 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, 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."

Additionally incidents are running at multiple other companies who are Snowflake customers where full databases have been taken — I have 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 utilizing Snowflake environments".

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. Now 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 take down 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 writeup, until it the URL for accessing it was also blocked. 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 Schrödinger's Blog.*

The threat actors here, from what I've managed to establish, are a teen crimeware group who've been active publicly on Telegram for a while.

In other words, a bunch of kids did all this damage. When Kevin writes "Let's Recap" ...

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 use it. It allows you to do this with effectively no security.

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 customer's 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 the coming weeks and months.

And boy was Kevin prescient about that! He wrote this more than a month before the AT&T breach announcement. Then he writes – and this is so perfect:

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.

What you're sold versus what you get often don't align — 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 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."*

[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?]

Snowflake have incident response stood up, with Crowdstrike (and Mandiant involved). They say the cause of the malicious activity (read: 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;*

This is curious, since they're not saying from where a huge number of their customer's 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 would be? As Kevin wrote: "in the age of SaaS [Software as a Service], your providers will throw you under the bus to save themselves."

So what happens, essentially, is info stealers were used to gain access to Snowflake databases using their customer's stolen credentials, using the client name rapeflake (side note to threat actor over that name: really?).

Snowflake themselves fell into this trap, by both not using multi factor 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 isn't great.. as they can point out customers messed up, but they messed up too.

Kevin then digresses a bit about what he feels is the hugely important topic of "infostealers", writing:

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.

Mandiant themselves have this to say about infostealers this weekend:

> 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 and use them to compromise enterprises, steal data, and conduct extortion.

If you use Snowflake, you need to first of all enable multi-factor 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 has been using your data — you cannot rely on Snowflake doing this for you.

Infostealers are a significant problem — it has long since outpaced botnets etc in the real world — and the only real solution is robust multi-factor authentication (and ideally getting rid of passwords altogether by replacing them with secure authentication).

There are companies offering services where you can buy your own stolen credentials back, and then you can change user's passwords. 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.

Tightening authentication fixes the problem. Ask the Snowflake victims how they have fixed the problem — it's through robust MFA. 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 scale of the breach that the status quo hasn't worked. Secure authentication should not be optional. And they've got to be completely transparent about steps they are 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, they need to recall Recall or they will pour petrol onto the flames and make the infostealer problem far worse.

After Kevin posted his piece he added a follow-up:

> *people are pinging me to say there's more to this story than disclosed. I know. It will be a developing story and all eyes are on Snowflake.*

The Record captured the examined Hudson Rock's posting before it was taken down. Referring to what was originally posted, The Record wrote:

> *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.*
>
> *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 400 companies are 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" targeting some customer accounts. They said: "Research indicates that these types of attacks are performed with our customers' user credentials that were exposed through unrelated cyber threat activity. To date, we do not believe this activity is caused by any vulnerability, misconfiguration, or malicious activity within the Snowflake product."*
>
> *Snowflake acknowledged that a former employee's demo account was accessed through stolen credentials, but said it did not contain sensitive data. They also said there is no "pathway for customers' credentials to be accessed and exfiltrated from the Snowflake production environment."*

Right. We've heard that before and whether or not those saying it at the time believed it, it later turned out not to be true. Anyway, The Record finishes, writing:

> *The company's response appeared to be at odds with claims by Hudson Rock that an apparently legitimate .CSV file of stolen documents showed over 2,000 customer instances connected to Snowflake's Europe servers.*
>
> *The files also purportedly showed that a Snowflake employee was infected by infostealer malware last October. According to the threat actor, these were the credentials used to carry out the attack.*

So, Snowflake insists that it didn't happen that way, while spewing the non-sequitur that only their customers who were not using multi-factor authentication had all of their data stolen. But they are not providing any answer to the question of how someone obtained so many of their customers' single-factor authentication data. Only the attacker's claims make sense and match the facts that have been seen.