CrowdStruck

Description: What do we know about how the FBI broke into the smartphone of Trump's deceased would-be assassin? Cisco scored another very rare CVSS 10.0 for a serious remote authentication vulnerability. If you're affected you MUST update. Untrusted Entrust's plan for the future is revealed. Surprisingly, Google loses the anti-third-party cookie battle. Third-party cookies stay. More interesting experiences from GRC's weekly Security Now! podcast mailings. Now we know why the company named itself "Snowflake." A collection of interesting listener feedback follow-ups on recent discussions. And we learn what in, literally, the world happened to allow CrowdStrike to crash 8.5 million Windows gateways, servers, and workstations to cause the largest IT outage of all time.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-984.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-984-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. And of course the topic of the hour, the day, the week, the year probably, is the CrowdStrike incident. Steve breaks it down, tells us what happened. Of course, some of this is speculation because we don't know all the details. But he will give you more details than anyone else. This is the place for the CrowdStrike story, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 984, recorded Tuesday, July 23rd, 2024: CrowdStruck.

It's time for Security Now!, the show where we cover your security, your privacy, and how things went awry last Thursday and Friday with this guy right here, Steve Gibson of GRC.com, our security guru. Hi, Steve.

Steve Gibson: Leo, it's great to be with you for the podcast that will surprise no one. I was using the phrase "This podcast wrote itself."

Leo: Yeah. But I think there's been a lot of interest from our audience about what your take is, what your explanations are, and all of that.

Steve: I actually have some that no one has read anywhere else.

Leo: Good. Look forward to that.

Steve: So, yeah, I think it's going to be interesting. So of course I titled this podcast "CrowdStruck." And the subtitle is "The Legend of Channel File 291."

Leo: Yes, mm-hmm.

Steve: So, yes, we're going to have...

Leo: A name, by the way, that IT professionals everywhere are learning to hate as they go from machine to machine, deleting them one by one.

Steve: Oh, my god, yes. And, well, there's so much to talk about because, like, why was that necessary? Why couldn't Windows restore itself?

Leo: Yeah, lot of people said that, yeah.

Steve: Why isn't there, like, some escape? And of course the real ultimate biggest issue of all is how could CrowdStrike have ever allowed this to happen?

Leo: Right.

Steve: I mean, like, what could possibly explain how this got out into the world without them knowing that, you know, like what was going on. So anyway, just this is going to be one for the record books, probably also in terms of total length because I ended up at, I think, 22 pages, which is four more than our normal - oh, no, 24 pages. Yikes. So, yeah, we have a lot to cover.

Leo: It's okay. We want this. We want it. Go, baby, go.

Steve: And believe it or not, Leo, something else actually happened last week.

Leo: No. No.

Steve: Besides the world basically coming to a grinding halt. So we're going to look at what we know about how the FBI broke into the smartphone belonging to Trump's deceased would-be assassin. Cisco managed to score another of a very rare CVSS 10.0s that you never want to score. It's a serious remote authentication vulnerability, as evidenced by having the maximum 10 out of 10 severity rating. So anyone affected absolutely must update. Also, we now know about the untrusted Entrust's plan for the future, how they plan to be moving forward once Chrome has said we're not going to be trusting anything you sign after Halloween. Also, oh, boy. Now, once upon a time I would have said the most tweeted. Now this was the most emailed to me note because all of our listeners know how I feel about this. Google has lost...

Leo: Yes.

Steve: ...the anti-third-party cookie battle.

Leo: Yes.

Steve: Oh, boy. So cookies are staying, and we'll have some things to talk about there. Also I'm going to share a few more interesting anecdotes from my weekly Security Now! podcast mailings, the experience from last week. And now I know about this week because 7,000 of our listeners received all of this two hours ago, you know, the breakdown of topics and the show notes and the Picture of the Week and so forth.

Leo: So you get feedback for the show even before you do the show.

Steve: Yeah, it happened.

Leo: That's great, yeah.

Steve: Yes, exactly.

Leo: That's good, you know, it's like a focus group kind of.

Steve: And one of those things was a guy writing from New South Wales, Australia, who actually wrote his letter to me, tweeted it, but I had already finished and produced the podcast before I finally went over to Twitter to tweet about the podcast, and I saw that there were some - anyway, he has something really cool to say about CrowdStrike which we'll get to next week because, hint hint, email is a little more quick getting to me.

Leo: Faster.

Steve: Yes.

Leo: Well, our friends in Australia were the first to bear the brunt of the CrowdStrike.

Steve: Yes, it was in the afternoon that his world as he knew it ended.

Leo: Poor guy.

Steve: Anyway, so I'm going to - also, we now know where the seemingly "flaky" name "Snowflake" came from, and why.

Leo: Ah. Oh.

Steve: And then I do have some listener feedback I want to share, following up on recent discussions. And then we're going to learn what in, literally, the world happened to allow CrowdStrike to take down 8.5 million Window's gateways, servers, and workstations to cause the largest IT outage of all time. And of course the reason we're not hearing the details is that you have to imagine that CrowdStrike's attorneys dropped the Cone of Silence over that, I mean, they probably just went over and yanked the phones out of the wall.

Leo: Yeah, yeah.

Steve: And said no one is saying anything.

Leo: No one's saying nothing.

Steve: And we do have a Picture of the Week for today's podcast.

Leo: Awesome.

Steve: Which I will talk about a little bit briefly as we get going here.

Leo: Very good. Well, a really big show coming up here in just a bit with Steve Gibson, and we will get right to it. All right, Steve. The Picture of the Week is not a joke this week. Far from it.

Steve: Not so funny this time.

Leo: Yeah.

Steve: I gave this snapshot the title "The simple memory pointer mistake that stalled the world." And I will be talking about this in gratifying detail at the end of the podcast. But what we see from this, now, this is the kind of crash dump that most people just like, what? You know, it's like, it's going to have no meaning whatsoever to almost anyone. But people who understand the machine architecture and dumps will see that this occurred, this crash occurred inside something called CS Agent, which doesn't take any stretch to know would stand for CrowdStrike Agent, and that there was a function which took two parameters. And this shows up like on the fifth or sixth line.

Leo: This is assembly language, which is why Steve knows what it means; right?

Steve: Correct, correct. Anyway, so the function was given a bad parameter - and I have a theory as to why, which we'll talk about - which caused it to attempt to load 32 bits from where there was no memory. And you can't do that.

Leo: Right.

Steve: It's one of the things that cause Windows to just give up. And I'll explain why Windows could not recover from this, why an application doing this is different from the kernel doing this, and all of that.

Leo: Good.

Steve: So, you know, basically this is a snapshot of the actual crash. And it was a simple memory pointer mistake, and it took down immediately 8.5 million machines, Windows operating systems.

Leo: To the point where they couldn't be rebooted. They just had to be fixed.

Steve: Yes, in a way that required - and that's one of the most expensive factors of this, right, is you had to visit, somebody had to visit...

Leo: Physically.

Steve: ...every single machine.

Leo: Lord above.

Steve: Oh, boy. So, you know, with the glare of what happened last week still looming over everything, it's a little difficult for us to focus upon anything else. And of course we're going to give this our full attention. But there was some other important news that emerged last week which should not be overshadowed.

In the wake of the failed attempted assassination of our ex-U.S. President Donald Trump during his recent campaign rally, the FBI has been attempting to learn all it can about the immediately deceased would-be assassin. You know, they obviously can't ask him for his password. So 20-year-old Thomas Matthew Crooks was using an Android phone that was password locked. And of course we've certainly been here before; haven't we? We all remember the San Bernardino mess.

Bloomberg reported that the FBI sought help from the Israeli digital intelligence company Cellebrite which, with offices conveniently located in nearby Quantico, Virginia, is known to provide smartphone unlocking technology to U.S. federal agencies. A significant percentage of their business is doing that. Sources familiar with the investigation, who requested anonymity, told Bloomberg that the FBI needed data from the phone to understand Crooks' motives for the shooting. Right, everyone wants to know, you know, why? Doesn't really matter, but still interesting.

So although the local FBI bureau in Pittsburgh already did have a current license for Cellebrite's smartphone cracking software, it was ineffective on Thomas Crooks' newest Samsung device. So undaunted, the FBI reached out to Cellebrite's nearby federal team, which is there for the purpose of collaborating with law enforcement when they've got some problems. Within hours of that, Cellebrite had provided the FBI with the additional

support they needed, including some newer, not-yet-released software. And 40 minutes after that, the FBI had Thomas's Samsung Android smartphone unlocked and open for detailed inspection of the shooter's social media, browsing, texting, whatever, history.

What's interesting is that in other, just it was coincidental really, reporting, it appears to be fortuitous for the FBI that Thomas was not using a later model Apple iOS device since some documents leaked from Cellebrite indicate its inability to unlock such devices. 9to5Mac picked up on this last Thursday, reporting under their headline "Cellebrite cannot unlock most iPhones running iOS 17.4 and later." They wrote: "Leaked documents reveal that Cellebrite cannot unlock iPhones running iOS 17.4 and later, at least as of the date of publication," which was April of this year. They said: "The company has confirmed that the documents are genuine. Cellebrite devices, which are widely used by law enforcement agencies, can crack most Android phones, though there are exceptions.

"Cellebrite's kit relies on discovering vulnerabilities discovered in iOS and Android, which Apple and Google of course then aim to fix, well, discover and resolve. Others also work to defeat the phone-cracking kit, which mostly secure messaging app Signal scored with a big win in 2021, when it managed to booby-trap iPhones to render the kit useless." And we covered that at the time. Back in 2022, 9to5Mac managed to obtain user documentation which iPhone models that kit at the time could not unlock.

Since then, and with this recent document discovery, it was 404 Media that grabbed updated docs, and these are the ones dated April of 2024. I got a look at the PDF. It was four panels of grid and explanation that used a lot of jargon that you'd have to have a glossary in order to untangle. So the reporting here is easier to understand. They said: "As of that date" - that is, April 2024 - "Cellebrite had not managed to crack iPhones running iOS 17.4 or later, which today is a very large percentage of iPhones."

9to5Mac said: "Additionally, the kit cannot currently break into most iPhones running iOS 17.1 to 17.3.1, though hardware vulnerabilities in the iPhone XR and 11 mean those are exceptions. The company (Cellebrite) appears to have worked out how to access other iPhones running those versions of iOS; however, as the table says this capability is 'coming soon'" - which means, but we don't know how to do it yet, so we don't really know if it's coming ever. The documents are titled "Cellebrite iOS Support Matrix" and "Cellebrite Android Support Matrix," respectively. An anonymous source recently sent the full PDFs to 404 Media, who said they obtained them from a Cellebrite customer.

For all locked iPhones able to run 17.4 or newer, the Cellebrite document says "In Research," meaning they cannot necessarily be unlocked with Cellebrite's tools today. We know from Apple that the majority of iPhones in use today are using iOS 17, though the company doesn't share breakdowns of the specific point numbers. That said, it's a safe bet that a high percentage were uncrackable by Cellebrite as of the date of the document.

A separate table of Android-cracking capabilities show that most of them are accessible by the kit, though the Google Pixel 6, 7, and 8 are exceptions if they were powered down at the time that they were obtained. That's because the cold-boot process blocks the exploit that's being used. But they can be accessed if powered up and locked. The same is true of Samsung phones running Android 6, but not those running later versions, indicating that Samsung's implementation of Android 7 managed to introduce a vulnerability which is still present all the way through Android 14, which Cellebrite knows about.

So anyway, The Verge summarized the state of play by writing simply: "Phone hacking companies are overstating their capabilities." And of course they have motivation to do so. The Verge noted that most newer phones are currently beyond the capabilities of these commercial phone hacking companies. This could mean that the phone vendors are

finally winning this battle by iterating over and constantly improving the security of their solutions. It could also mean that older phones are currently vulnerable because the companies have had more time with them and that these newer phones may similarly fall in the future. We can't really say.

I guess if I were a betting man I'd go short on the stock of the hacking companies, since I suspect their remaining days are numbered as the hardware finally becomes impregnable. But please don't take this as a stock tip. I'm not a betting man, and I would never encourage anyone else to be. CrowdStrike has just demonstrated that anything can happen. So you never know with computers and software.

As we know, CVEs having a CVSS score of 10.0 out of a possible 10.0 are vanishingly and blessedly rare. Unfortunately, last Wednesday, Cisco was forced to report and acknowledge one of their own. Ars Technica wrote: "On Wednesday, Cisco discovered a maximum-security vulnerability that allows remote threat actors with no authentication to change the password of any user, including those of administrators with accounts, on Cisco Smart Software Manager On-Prem devices." So that's the key phrase for any of our listeners. And, boy, I'm getting an education about just at what a high level our listeners are operating throughout their organizations. So I wouldn't be at all surprised if this means something to some of them. Cisco Smart Software Manager On-Prem devices. If you have one around, and it hasn't been patched since last Wednesday, hit pause on the podcast and go do that.

"The Cisco Smart Software Manager On-Prem resides inside the customer premises and provides a dashboard for managing licenses for all Cisco gear in use." So, you know, they created sort of a central management hub which on the one hand makes it very convenient for managing everything in your organization. On the other hand, if you happen to have a CVSS of 10.0 which allowed non-authenticated remote users to change passwords at will, that would be a problem. That concentration of power which we keep seeing now, this seems to be a repeating theme; right? Over and over, yes, it's convenient. But, boy, when it gets hit, it makes the pain much worse. "So it's used by customers who can't or don't want to manage licenses in the cloud, which is the more common approach.

"So in their bulletin, Cisco warns that the product contains a vulnerability that allows hackers to change any account's password. The severity of the vulnerability" - this is Ars speaking - "tracked as CVE-2024-20419, is rated 10," they write, "the maximum score." The Cisco bulletin stated: "This vulnerability is due to improper implementation of the password-change process." Okay, that seems kind of obvious. "An attacker could exploit this vulnerability by sending crafted HTTP requests" - in other words web request - "to an affected device. A successful exploit could allow an attacker to access the web UI or API with the privileges of the compromised user." There are no workarounds available to mitigate the threat. Other than, you know, pulling its plug, but that would probably be a problem, too.

"So it's unclear precisely what an attacker can do after gaining administrative control over the device. One possibility is that the web UI and API, which the attacker gains administrative control over, could make it possible to pivot to other Cisco devices connected to the same network and, from there, steal data, encrypt files, you know, get up to all the bad stuff that the bad guys do these days. Cisco reps did not immediately respond to email queries from Ars Technica." And they finished by noting that the post would be updated if a response were to come later. So if there's an update, you definitely want to apply it. So again, ultra rare, 10.0. If you know anybody who has one of these, make sure they update it since last Wednesday.

Okay. So in following up on one of our past big points of coverage, several weeks ago, as we all know, after a great deal of handwringing and teeth gnashing on the part of those

who run the collective known as the CA/Browser forum, Google decided that they could no longer in good conscience afford to have their Chrome web browser honor and trust certificates signed by Entrust moving forward. This was not done, as had been done in the past, due to any egregious, horrific certificate mis-issuance event, but rather to send a very strong and clear message, not only to Entrust, but to the entire community of certificate authorities that there would actually be consequences if they did not live up to the operational and behavioral commitments that they themselves had previously agreed to.

Among many other aspects of this, it was not fair for Entrust to be allowed to leave their own misissued certificates in place and thus saving face with their customers, while other CAs were playing by the self-imposed rules by going to the cost and inconvenience of acknowledging, revoking, and reissuing any mistakes that they may have made. So Google's move was, and it was meant to be, a very clear demonstration that this game would not tolerate any multiyear endemic cheating.

The week after we covered this historic event, Todd Wilkinson, Entrust's president and CEO, formally apologized and said once again, after the industry had lost count of the number of previous times Entrust had said this, that they were really, truly, and seriously this time going to do better. And I suspect that this time they probably will. But that left us with the question, what would Entrust do in the meantime? When we were talking about this we explored various paths. We're back here talking about this today because Todd is present with the answer to that question.

He signed the following newsflash from them, writing: "To our TLS customers." He said: "I would like to thank you for your patience as we diligently work to ensure that you will continue to receive uninterrupted public TLS certificate services through Entrust. Today we are ready to share our go-forward plans. First, as you likely know," he says, "Google said that Chrome will no longer accept Entrust public TLS certificates issued after October 31st, 2024. Entrust TLS certificates issued prior to October 31st will continue to be accepted through their expiration date.

"Entrust is committed to returning to the Chrome Root Store and will keep you informed of developments. We've identified the steps to address Google's decision. We continue to execute our improvement plans and are working closely with the browser community in discussions on our path forward. In the meantime, after October 31st, 2024, you can continue to request public certificates and receive certificate services directly from Entrust. Here is how this will work."

And we have three bullet points: "First, continue to order certificates as you have been, under the same pricing model and service-level agreements (SLAs). Second, rely on Entrust for certificate lifecycle management, verification, support, and professional services, as we plan to serve as the Registration Authority for these certificates. And finally, we will deliver public TLS certificates issued by a CA partner that meets the requirements of the CA/Browser Forum and Entrust." And he finishes: "Today we can share that SSL.com is now an Entrust CA partner. SSL.com is a global CA founded in 2002 with full browser ubiquity. They are used by businesses and governments in over 180 countries to protect internal networks, customer communications, e-commerce platforms, and web services, and we are pleased to partner with them to meet your needs."

And he finishes: "To build resilience into your organization, we recommend that you take inventory and renew your Entrust certificates prior to October 31st, 2024. These certificates will be trusted through their expiration date, up to 398 days." By which time, I'm sure, he's hoping this will no longer be necessary. Anyway, he finishes: "You can renew your certificates through your certificate lifecycle management solution, automation tool, or the Entrust Certificate Services Portal. Signed, Todd."

Okay. So that answers that question. Quote: "We will deliver public TLS certificates issued by a CA partner that meets the requirements of the CA/Browser Forum and Entrust." So it does not appear that any other CA is going to be allowing Entrust to ride on their coattails by signing a new Entrust intermediate certificate that has the power to, in turn, sign web server identity end certificates. Instead, Entrust found SSL.com, an even smaller CA than them, who is in good standing, from whom they will purchase and resell web server TLS identity certificates.

The best estimates I've been able to find on the web are that Entrust does indeed, as we noted previously, have about 0.1% of the total website server business. SSL.com appears to have about half of that at 0.05%. So this deal represents something of a windfall for SSL.com. Entrust will presumably use SSL.com's certificate issuing machinery in return for paying SSL.com for every certificate Entrust issues under their auspices.

So it's a win-win for the time being, but this does also feel like a temporary backstop solution for Entrust. It feels as though Entrust does indeed plan to work to rehabilitate itself in the eyes of the CA/Browser community to then have Chrome and any other browsers that may be planning to follow Chrome's lead restore their trust in Entrust's operations and integrity. So though Entrust will be losing out on some fraction of their overall certificate revenue, they will likely be able to retain the customer relationships they've built and will someday be able to again issue certificates under their own name.

So, and of course you can see, based on what Todd is saying, he is saying, if you use Entrust before Halloween to reissue a certificate you have, that will span the next 398 days. And they're hoping to be rejuvenated by that point. So maybe they won't even need to fall back on SSL.com. But certainly there will be customers who will be, like, aren't listening to the podcast, are completely clueless about any of this happening, who will be coming back to Entrust later in the year or any time next year as their previously issued Entrust certificate is getting ready to expire. Entrust is saying, that's fine, we're not losing you. You can still, you know, everyone's pretending that we're still issuing your certificate. You use our portal, you use our UI, but actually the certificate will be coming from SSL.com in return for us giving them some piece of the action in order to perform that service for us. So that appears to be what they're up to. And Leo, we're about 30 minutes in.

Leo: Yeah, let's take a...

Steve: So this would be a good time to take a break, and I'm going to take a sip of coffee to whet my whistle.

Leo: We'll take a little time out and be back with more. You know what, you found a lot of other stuff to talk about. It's not all CrowdStruck. Absolutely not. In fact, Google's cookies coming up in just a little bit.

Steve: Ooh.

Leo: Ooh, boy.

Steve: Sadly.

Leo: Ooh, boy. All right, Steve. What's all this about cookies? You really liked this whole Topics thing that they were going to do; right?

Steve: Well, it made sense. I understood it.

Leo: Yeah.

Steve: It would have worked.

Leo: It was privacy forward, yeah.

Steve: Oh, absolutely. So just sort of to recap. Because web servers and web browsers operate query by query, one query at a time, long ago Mozilla designed a simple add-on called a cookie. A website's server could easily give a browser one of these unique cookies, just a meaningless string of data to the browser. But it would subsequently return that, thus identifying itself to the website for all subsequent activities. This simple solution enabled the concept of being "logged on" to a website, which never existed before. And this was the same way that users were previously able to log into other online services. So it was a breakthrough. But that's what it was meant for.

As this podcast's longtime listeners know, I've always been very annoyed by the abuse of this simple cookie technology by third parties, since cookies were purely and expressly intended to be used as a means for maintaining logged-on session state, and nothing more. But the advent of third-party advertisers whose ads poked out onto tens of thousands of websites all over the world meant that their third-party cookies could be used to track and thus profile users as they moved across the web.

And as you said, Leo, for this reason I've been very excited and hopeful about actually all of Google's sincerely repeated attempts to design a workable alternative, which, you know, they would, they've said, once they had that, completely eliminate all support for third-party cookies. Now, after that, the web would, in my opinion, finally be operating the way Mozilla originally intended, without cookie abuse, while still offering advertisers the feedback about the visitors to websites who were viewing their ads that they wanted.

The only problem was the achievement of this goal would also collapse the entire Internet tracking, profiling, and data aggregation industry. And for that reason it appears that Google has failed in their quest, and that the tracking and profiling industry has won. Yesterday, Monday, July 22nd, Google's VP of the Privacy Sandbox project effectively admitted defeat.

Anthony Chavez's posting was titled "A new path for Privacy Sandbox on the web." Well, you know, the path they were on was the right path, so a new path is not going to be any righter. Understanding what Anthony is really saying here requires a great deal of reading between the lines, though we've focused enough on this in the past that I think I know what actually happened.

Anyway, first, here's what he wrote. He said: "We developed the Privacy Sandbox with the goal of finding innovative solutions that meaningfully improve online privacy while preserving an ad-supported Internet that supports a vibrant ecosystem of publishers, connects businesses with customers, and offers all of us free access to a wide range of content." Right.

He says: "Throughout this process, we've received feedback" - I bet you have - "from a wide variety of stakeholders, including regulators like the UK's Competition and Markets Authority (CMA)" - that we were talking about not long ago - "and Information Commissioner's Office (ICO), publishers, web developers, and standards groups, civil society, and participants in the advertising industry. This feedback has helped us craft solutions that aim to support a competitive and thriving marketplace that works for publishers and advertisers, and encourage the adoption of privacy-enhancing technologies.

"Early testing from ad tech companies, including Google, has indicated that the Privacy Sandbox APIs have the potential to achieve these outcomes. And we expect that overall performance using Privacy Sandbox APIs will improve over time as industry adoption increases. At the same time, we recognize this transition requires significant work by many participants and will have an impact on publishers, advertisers, and everyone involved in online advertising. In light of this, we are proposing an updated approach that elevates user choice." Now, that's called putting a good face on the problem.

They said: "Instead of deprecating third-party cookies, we would introduce a new experience in Chrome that lets people make an informed choice that applies across their web browsing, and they'd be able to adjust that choice at any time. We're discussing this new path with regulators and will engage with the industry as we roll this out." They said: "As this moves forward, it remains important for developers to have privacy-preserving alternatives. We'll continue to make the Privacy Sandbox APIs available and invest in them to further improve privacy and utility. We also intend to offer additional privacy controls, so we plan to introduce IP Protection into Chrome's Incognito mode.

"We're grateful to all the organizations and individuals who have worked with us over the last four years to develop, test, and adopt the Privacy Sandbox. And as we finalize this approach, we'll continue to consult with the CMA, ICO, and other regulators globally. We look forward to continued collaboration with the ecosystem on the next phase of the journey to a more private web." And of course the problem is none of the other stakeholders want a more private web.

So third-party cookies will remain in Chrome, and it really appears unlikely that technologies such as the privacy-preserving Topics will gain any foothold since the advertising, tracking, profiling, and data aggregating industries want everything they can get their hands on. And they appear to have won this battle by crying to European regulators that it's no fair for Google to take this away from them.

It'll be interesting to see how Google's capitulation manifests in this user-interface change they're talking about and whether websites begin insisting that their users enable what is going to be called, you know, whatever they end up calling it across their site's content. You know, like, oh, you seem to have this turned off. If you want to use this content, please turn it on. Oh, and while you're at it, what's a good email address for you?

Leo: But most people who certainly listen to this show turn off third-party cookies. Is that setting honored in all browsers?

Steve: Yes.

Leo: Okay.

Steve: Yes. And I can vouch for that because this, you know, this has been a hobbyhorse of mine for quite a while. If you were to google "GRC cookie forensics," it takes you to a series of pages I created years ago. And actually I'm seeing that third-party cookie behavior is down among GRC.com's visitors. Yes, you're showing it now. That cookie forensics page is actually able to verify...

Leo: Wow.

Steve: ...the percentage. Yeah, so that used to be up at about 80%. And now as we can see it says 17.8% of all GRC visitors have persistent third-party cookies enabled. And that's of 21,767 unique GRC visitors.

Leo: So people trying to get back on?

Steve: No, no, no. Scroll down and look at the Apple number.

Leo: Ah.

Steve: Because Apple is the only one who's always had this off by default. So again, this shows the tyranny of the default. Apple Safari visitors are down below 2% of them.

Leo: Okay.

Steve: But it's on by default, and that says that a lot of people are turning it off.

Leo: And that's because they listen to this show. They know to turn it off. That's good.

Steve: Yes. And if you scroll down to the cookie forensics, there is a - let's see, which one? There, number three, it'll actually do it. That just tested your browser to look at your third-party cookie handling across the board. And yours is completely locked down.

Leo: And of course, because I listen to you, I have always turned off third-party cookies.

Steve: Yup.

Leo: Good job. So you know a lot about this.

Steve: Yes, this has been a focus of mine because it just really, you know, irks me that this has been so abused. And what's interesting is back when I originally designed it, that forensics test, you'll notice it has eight different types of cookies it looks for of each type. It's because browsers...

Leo: They're sneaky.

Steve: So there's page, CSS, script, embedded, image, icon, iframe, and object. Browsers used to be broken. They were actually, even if you turned them off, some of them would still be on. So this allowed us to profile whether browsers were even operating correctly. And back in those early IE6 days, they weren't.

Leo: Well, the other point I think it's really good to take from this page is that first-party cookies are okay. That is a necessary system on the Internet. You don't want to turn off all cookies.

Steve: You can't log on. You can't log onto a website without first-party cookies.

Leo: And it's a convenience. It's the third-party cookies that are a problem. And it's unfortunate because I think these cookie banners from the UK and the EU have kind of taught people all cookies are bad. But that's not the case.

Steve: No.

Leo: Grr.

Steve: Yup. Anyway, so what I think must have happened, Leo, is because we saw the writing on the wall, it was the EU and their regulators that were under pressure from...

Leo: Advertisers, yeah.

Steve: Yes, the advertisers and the trackers and the data aggregators. It's like, hey, this is our business. You can't, you know, Google just can't come along and take away our business. It's like, unfortunately, their business should not be ours. And these have been, you know, mixed up. So I don't know how to really read what Google's thoughts are in the future. You know, they've got an API that requires, in order for it to work, they had to force everyone to use it. Otherwise no one's going to use it. You know, no one's going to make a change. So I think we're stuck with third-party cookies.

Leo: So again, if you google "GRC cookie forensics," you can see this page, free page on Steve's site. It's a really useful and informational page that I wish all of these regulators would read, to be honest with you. Oh, well.

Steve: They don't want it to be there. They just want to, you know, skulk around in the dark and aggregate data about us. Okay. So I thought that our listeners would enjoy learning what I've been learning from the exercise of sending email in this era of hyper-vigilant anti-spam and anti-viral email protection measures. Last Tuesday I found three causes for the trouble that I mentioned in getting the email to our listeners.

First, it turns out by sheer coincidence the thumbnail image of the Picture of the Week was indeed triggering a false positive detection from some AV scanners. Later that evening, I removed the thumbnail from the email and re-sent the email to the 83 subscribers who had not received it due to an AV rejection false positive, and that time only three of them bounced. So it was indeed just a coincidental thumbnail image. There's nothing I can do about it. And this week's thumbnail had no problem.

The second issue was, after carefully examining the feedback that I was receiving from some of the AV tools, I saw that some of them were complaining about the email containing a banned URL. The exact quote was: "Contains a URL listed in the URIBL blacklist." And guess what the URL was? Polyfill.io. I used that phrase in the email; and I didn't, like, put square brackets around the dot as I should have done. So, yep, you got me on that one. From now on I'll make sure that any dangerous URLs are rendered non-dangerous in any email that I send. But, so, yeah, kind of props for the AV guys for catching it, even though it was a pain.

The third and final discovery was a complaint about invisible text in the email. And so I stared at my code. I handwrote the HTML. Like, what? There's no invisible text. But it turns out I had taken an innocuous-looking line from several discussions about composing email for better viewing on a wider range of devices. It was an HTML line with its "display" style set to none and a font point size of 0. I don't know why...

Leo: Oh, it's a tracking pixel. That's a tracking pixel.

Steve: Yes. Well, no. But no, no, there was no URL.

Leo: It doesn't phone home.

Steve: There was no text. There was no URL. But what it did was it basically triggered another false positive. And I removed it from this week's mailing, and I didn't have any trouble at all. I should note, however, that it was mostly our listeners using Hover's email hosting service that were rejecting and bouncing last week's Security Now! email back to me. The bounce message was: "5.7.1: Message blocked due to the very low reputation of the sending IP." And it's like, okay, well, yeah, true. I'm just getting started here at GRC, sending these kinds of mails. So that was expected, and it doesn't make me love Hover any less. They're still my beloved domain...

Leo: Well, and you said it was a ClamAV, which that makes sense that they would use as an antivirus scan, it's a free open source program.

Steve: Yes, although this is - I'm not sure where the reputation of the sending IP comes from.

Leo: Oh, right.

Steve: Yeah. I did confirm with one of our listeners, a Paul Sylvester, whom I exchanged email with, he added securitynow@grc.com and mail-manager@gmail.com to his allow list. There is an allow list at Hover in the webmail system.

Leo: So you can do that.

Steve: Yes. And today I only got four bounces back from Hover. Everybody else apparently looked into it and fixed it.

Leo: Do you use DKIM and SPF? Do you use the authentication?

Steve: Oh, yeah. You don't even get off the ground unless you've got SPF, DKIM and SPF. I have them all. The problem is, in the same way that I'm signing all of my SpinRite EXEs, yet Windows Defender still sometimes complains. Similarly, even though my email is signed, if you don't have reputation, reputation matters. And so GRC.com, you know, I've never been doing anything like this kind, I mean, even 7,000 pieces of email. This morning 7,000 pieces of email went out to Security Now! listeners, and it almost went perfectly.

Leo: Yeah.

Steve: So it was much better than last week, and I imagine it'll only keep getting better in the future.

Leo: Well, and you build reputation as you use it.

Steve: Over time.

Leo: Yeah, over time.

Steve: Right, yeah.

Jeff Garretson in Yakima, Washington, said: "One of Snowflake's primary target markets is data warehouse applications. Traditionally, data warehouse databases are organized as a 'star schema,' with a central 'fact table' linked to multiple 'dimension tables' that provide context. A variation is when one or more dimensions have enough internal complexity that it makes sense to break some attributes out into sub-dimensions. Then the star schema diagram starts looking more complex, more like a snowflake. So a 'snowflake schema' is a more general case of a star schema.'" And he said: "Hope that helps. Love the show."

Leo: Okay.

Steve: And Jeff, thank you. That is indeed where the Snowflake name came from. I also got a kick out of a much more playful posting. At the moment, of course, as we know, Snowflake has a problem because some 350 of their clients had all of their data stolen from them, which does not make them happy.

But 10 years ago, back in 2014, a Snowflaker named Marcin Zukowski, who is a co-founder and VP of engineering at Snowflake, posted the following to Snowflake's blog. He

wrote: "One of the questions that we get the most is, 'Why did you decide to name the company Snowflake?'" He says: "I'm sure our marketing department has their opinion of what we should say, but let me give you the real story. The name Snowflake just fits us in several ways." And he has four bullet points.

"First, Snowflakes are 'born in the cloud.' For a data warehouse built from the ground up for the cloud, that's very important. Second, we love the snow." He says: "We love the snow. Most of our founding team, and even our first investor, love to spend time up in the mountains in the winter. They even convinced me to try skiing, and took me on a black run my first day. And third," he said, "each Snowflake is unique.

"One of the really cool things about our architecture is that it lets you have as many 'virtual warehouses' as you need, all in one system, each of which has exactly the right resources to fit the unique needs of each set of your users and workloads. And conveniently, 'Snowflake' happens to have a meaning in the world of data warehousing A data warehouse schema organized as multiple dimension tables surrounding a set of fact tables is one of the data architectures that we can support." So now we know.

Leo: Maybe more than we wanted to know, actually.

Steve: Well, we were making fun of them last week, so that's only fair.

Leo: That's true, yes.

Steve: To explain where they came up with as funky a name.

Leo: It's not completely out of the blue.

Steve: No.

Leo: Well, it is, it's out of the cloud.

Steve: A listener requesting anonymity shared his experience following the recent CDK Global dealership outage. He said: "Hi, Steve. I'm a software developer, and I develop an interface between my company's software and CDK dealerships. Our software sends tens of thousands of transactions to CDK daily. Our software tried to post many, many thousands of records during the outage; and since they were down, all transactions failed. The article you mentioned last week about the accounting office needing to deal with the mess is spot on.

"As I'm listening to the podcast about this mess a couple of days after it came out, I'm in the middle of crafting SQL scripts to 'fake out' the system to make it think that items that the dealership accounting offices had to manually handle were already posted across to CDK so that they would not double book them. That's basically anything that happened in June because the accounting offices had to close the books at the end of June.

"CDK cut all third-party interface access during their restoration. Our interface access was finally restored a few days ago. However, as part of the process of restoring our

interface access, CDK changed our interface credentials which had remained the same for 15 years. Yes, 15 years." So regards from an anonymous listener.

John Meuser, writing about the Polyfill.io mess and the use of resource hashes, wrote: "One thing I feel you should have mentioned with the SRI system" - you know, that's the system for tracking hashes of downloaded resources - "is that this does invalidate one of the reasons a web developer might use externally hosted resources. Suppose there's a vulnerability found in one of the external libraries? In that case, the website developer will have to update their URL and the hash before the vulnerability is fixed for their site. If they blindly pull the latest compatible version, they will always have the latest bug fixes. There will always be a difficult balance to be struck between convenience and security."

And John's right, of course. As I noted last week, the only way to safely verify a downloaded resource against its known hash is if that resource never changes. That can be assured only by specifying the resource's exact version number. But as John notes, that also means that the web pages using that "pinned" version release will not be able to automatically receive the benefits of the version being moved forward as bugs are found and fixed. So choose your poison. Either tolerate a bug that may later be discovered until you're able to update the version that your website is pulling, or go for the latest automatically and hope that you never download malware by mistake, as the Polyfill.io debacle showed was possible.

Simon, an Aussie in the UK, as he describes himself, he says: "I suggest MITM now stand for Miscreant in the Middle." And I like that one a lot. We don't need to change the abbreviation away from MITM, just the first "M" now stands for Miscreant. Thank you, Simon.

Ryan Frederick wrote: "You said on this week's show that you will make a Copilot+ blocker app in assembly, if Microsoft ever releases it. At this time, the only Copilot+ certified PCs are ARM, while you're an x86 assembly developer. That said, if anyone can learn ARM assembly in a week and release a patching program, it's you."

Okay. So first, Ryan makes a very good point. However, we know that there's no way ARM-based Windows machines will not also be able to emulate Intel x86 family instructions to run all existing Intel-based apps. We know they'll have a problem with drivers because that's - because they're down in the kernel. Apps they're going to be able to run. And Microsoft has indicated that Copilot+ with Recall will be coming to Intel platforms just as soon as they're able to make it happen. So I'm pretty sure that my style of app development will not be threatened. And it will be a long time, as in decades, before the population of ARM-based Windows desktop becomes important or significant or threatens Intel. So I think we're probably going to be okay. Which is good because I don't think I would develop an app in ARM assembly for Windows, if that's what it took.

Kris Quinby in Riverdale said: "I'm a week behind the podcast, finishing an audiobook. I'm also late in sending this feedback so it may have already been sent to you by hundreds of listeners." No. "In Episode 982" - which was just two weeks ago, so he's not that far behind - "you talked about a Linux Daemon that would monitor logs and modify firewall rules to block IP addresses that made unwanted connection attempts to the computer. You stated that you could not think of any reason why that should not be in place on every computer that accepts incoming connections."

So he says: "One downside is that active blocking can be used to create a denial of service condition. If the attacker notices the active blocking, they can spoof the source addresses for connection attempts to make the server start blocking all inbound connections. Since the connection does not need to be 'real' [he has in quotes], the TCP connection handshake is not required. There can be a balance where the new firewall

rules can have a time limit before they're removed. But if left to 'block forever,' a server can effectively be disconnected from the Internet. Signed, Kris."

Okay. So generically, I completely agree with Kris. He's absolutely correct that blocking incoming connections by IP opens up the possibility of creating deliberate denial of service attacks by deliberately filling up the block list of IP addresses. But I'm a little unsure what he meant when he wrote "Since the connection does not need to be 'real,' the TCP connection handshake is not required," because I believe it is. TCP-based services will not consider a client to be connected until the three-way handshake has been acknowledged. It's true that the client can provide data along with the final ACK in its reply to the server's SYN/ACK, but those roundtrips definitely do need to occur before the server's TCP/IP stack decides that it has a valid connection.

And of course you can't get roundtrips back and forth if you're spoofing your source IP because the acknowledgment packet will go off to the IP that you're spoofing, not back to you, so you're unable to complete the handshake. So while UDP services, which do not have a TCP three-way handshake, could definitely be spoofed to create such an attack, TCP-based services such as SSH, which is what we were talking about, the OpenSSH flaw fortunately cannot be spoofed. So blocking based upon authentication failures would be spoof proof for them.

And Leo, this has brought us to the one-hour point, and we are now ready to talk about CrowdStruck.

Leo: Oh, wow.

Steve: Let's do our third break, and then we're going to plow into...

Leo: Get into it, yes.

Steve: ...exactly what happened.

Leo: I know everybody's really interested in your take on this, so I can't wait. All right. Back to CrowdStruck. I'm very curious what you have to say.

Steve: The Legend of Channel File 291. So I start this in the show notes with a picture from a listener. This shows the blue screens of death at the deserted Delta terminal of the Seattle Tacoma Airport, which was taken Friday morning by a listener who has been listening since Episode 1. And we see three screens. It looks like maybe there's someone back in the distance there sort of behind one of the screens facing away from us. But otherwise, there's nobody in line.

Leo: It's deserted.

Steve: There's an empty wheelchair. It's...

Leo: 4,000 flights canceled. 4,000 flights canceled. For Delta alone. By the way, Paul Thurrott, being a little bit of a pedant, says that's not the Blue Screen of Death,

that's a recovery screen. But you know what? It's a Blue Screen of Death. It's a Blue Screen of Death.

Steve: Okay, Paul. Point taken. So it's fortunate that GRC's incoming email system was already in place and ready for this CrowdStrike event.

Leo: I bet you got the mail. Whew.

Steve: Oh, did I. And I'm going to share some, right from firsthand accounts from the field, because it enabled a number of our listeners to immediately write last week to send some interesting and insightful feedback.

Leo: A lot of our listeners I'm sure have very sore feet going from machine to machine all weekend.

Steve: Oh.

Leo: Holy cow.

Steve: In one case, 20,000 workstations.

Leo: Oh, oh.

Steve: And these people don't hang out on Twitter, so email was the right medium for them. Brian Tillman wrote: "I can't wait to hear your comments next week about the current cloud outages happening today. My wife went to a medical lab this morning for a blood test and was turned away because the facility cannot access its data storage."

Leo: Wow.

Steve: Another listener wrote: "Good morning. I'm new to this group, only been" - I love this. "I'm new to this group, only been listening for the last eight years."

Leo: Oh, a newbie.

Steve: That's right. You're going to have to go back and catch up, my friend. He says: "I'm sure CrowdStrike will be part of next week's topics." Uh-huh. He said: "I would love to hear your take on what and how this happened. I'm still up from yesterday, fixing our servers and end-users' computers. I work for a large hospital in central California, and this has just devastated us. We have fixed hundreds of our critical servers by removing the latest file pushed by CrowdStrike, and are slowly restoring services back to our end-users and community. Thank you for all you do keeping us informed and educated of issues like this. Looking forward to 999 and beyond."

Leo: Oh, I like - that could be our new slogan, "999 and Beyond." I like it.

Steve: Doot da doo.

Leo: That's it.

Steve: Tom Jenkins posted to GRC's newsgroup: "CrowdStrike is a zero-day defense software, so delaying updates puts the network at risk. I don't know how they managed to release this update with no one testing it. Seems obvious at this point even casual testing should have shown issues." And of course Tom raises the billion-dollar, and I'm not probably exaggerating, the billion-dollar question: How could this have happened? And we'll be spending some time on that in a few minutes. But I want to first paint some pictures of what our listeners experienced firsthand.

Tom's posting finished with: "We had over 100 servers and about 500 workstations offline in this event, and recovery was painful. Their fix required the stations to be up. Unfortunately, the bad ones were in a boot loop that, for recovery, required manual entry of individual machine BitLocker keys to apply the fix." And of course that was often a problem because machines that were protected by BitLocker needed to have the recovery keys present in order for their maintainers to be able to get to the file system, even after being booted into Safe Mode, because Safe Mode's not a workaround for BitLocker.

Seamus Marrinan, who works for a major corporation which he asked me to keep anonymous, although I asked for permission and he told me who it is, but asked for anonymity there. He said: "For us, the issue started at about 12:45 a.m. Eastern time. We were responding to the issue" - and get a load of this response and the way his team operated. So the issued started for him at 12:45 a.m. Eastern time. He said: "We were responding to the issue by 12:55" - 10 minutes later - "and had confirmed by 1:05 a.m. that it was a global level event, and communicated that we thought it was related to CrowdStrike. We mobilized our team and had extra resources on site by 1:30 a.m.," so 25 minutes later.

"The order of recovery we followed were the servers, production systems, our virtual environment, and finally the individual PCs. In all, there were about 500 individually affected systems across a 1,500-acre campus. We were able to get to 95% recovery before our normal office hours started, and we were back to normal by 10 am." Okay, now, I am quite impressed by the performance of Seamus's team. To be back up and running by 10:00 a.m. the day of...

Leo: That's amazing.

Steve: ...after 500 machines were taken down across a 1,500-acre campus, taken down in the middle of the night, is truly impressive. And I would imagine that whomever his team reports to is likely aware that they had a world-class response to a global-scale event since, for example, another of our listeners in Arizona was walking his dog in a mall because it's too hot to walk pets outside during the day in Arizona. He took and sent photos of the sign on Dick's Sporting Goods the following day, on Saturday, stating that they were closed due to a data outage. So it took many other companies much longer to recover.

A listener named Mark Hull shared this. He said: "Steve, thanks for all you do for the security community. I'm a proud SpinRite owner and have been an IT consultant since the days of DOS 3 and Netware 2.x."

Leo: Wow.

Steve: Uh-huh. He said: "I do a lot of work in enterprise security, have managed CrowdStrike, write code, and do lots of work with SCCM," he says, "(MS endpoint management), as well as custom automation. So I feel I have a good viewpoint on the CrowdStrike disaster." He says: "CrowdStrike is designed to prevent malware, and by doing so provide high availability to all our servers and endpoints. The fact that their software may be responsible for one of the largest global outages is completely unacceptable. As you have said many times, mistakes happen. But this kind of issue represents a global company's complete lack of procedures, policies, and design that could easily prevent such a thing from happening." Now, of course, this is, what do they call it, something quarterbacking. Monday night?

Leo: Yeah, yeah, Monday night quarterback, yeah, yeah.

Steve: Okay.

Leo: 20/20 hindsight, yeah.

Steve: Okay. And I do have some explanations for this which we'll get to. Anyway, he said: "Given that CrowdStrike is continually updated to help defend against" - and I should just say no one's disagreeing with him, and Congress will be finding out before long, you know, what happened here. But he said: "Given that CrowdStrike is continually updated to help defend against an ever-changing list of attacks, the concept of protecting their customers from exactly this type of issue should be core to their design. Working in automation, the rule is that you always have a pilot group to send out software before you send it to everyone."

He says: "I work with organizations with easily over 100,000 users. If you don't follow these rules, you eventually live with the impact. In the old days, companies would have a testing lab of all kinds of different hardware and OS builds where they could test before sending anything out to production. This would have easily caught the issue," he says. "Now it seems that corporations have eliminated this idea since this is not a revenue-generating entity. They should research opportunity cost." He says: "With the onset of virtualization, I would argue the cost of this approach continues to decrease." And again, at this point this is speculation because we don't understand how this happened.

But, he says: "Since it appears this was not being done, another software design approach would be to trickle out the update, then have the code report back metrics from the machines that received the update at some set interval. For instance, every five, 10, 30 minutes the endpoints could send a few packets with some minor reporting details, such as CPU utilization, disk utilization, memory utilization. Then, if CrowdStrike pushed an update, and the first thousand machines never reported back after five minutes, there would be some automated process to suspend that update and send emails out to the testing team. In the case of endpoints that check back every 10 minutes, you could set a counter," and blah blah blah.

Anyway, he goes on to explain, you know, the sorts of things that make sense for means of preventing this from happening. And, yes, I agree with him completely. From a theoretical standpoint, there are all kinds of ways to prevent this. And again, as I said, we'll wrap up by looking at some of that in detail.

Samuel Gordon-Stewart in Canberra, Australia, wrote. He said: "Here in Australia it was mid-afternoon on a Friday. Most broadcast media suffered major outages, limiting their ability to broadcast news, or anything else for that matter. Sky News Australia had to resort to taking a feed of Fox News as they couldn't even operate the studio lights. They eventually got back on air in a limited capacity from a small control room in Parliament House. The national government-funded broadcaster ABC had to run national news instead of their usual state-based news services and couldn't play any pre-recorded content, so reporters had to read their reports live to camera. A lot of radio stations were still unable to broadcast even Friday night.

"Supermarkets had their registers go down. One of the big supermarkets near me had half their registers offline. A department store nearby had only one register working. Train services were halted as the radio systems were all computerized. Airports ground to a halt. Half a dozen banks went offline. Telecommunication companies had outages. Many hospitals reverted to paper forms. A lot of state government systems seemed to be affected, but the federal government seemed less impacted. And who knows how long it will take for IT departments to be able to physically access PCs which won't boot so they can implement the fix. As you would say, Steve, about allowing a third party unilaterally updating kernel drivers worldwide whenever they want, 'What could possibly go wrong?'"

After I thanked Samuel for his note, he replied with a bit more, writing: "In my own workplace, we're offline until Monday. I think we got lucky because our network gateway was the first to take the update and failed before anything else..."

Leo: How clever.

Steve: "...had a chance to receive the update." I love that.

Leo: So it took them offline, but fortunately it stopped the update for everybody else. That's good. I like it.

Steve: Yup. He said: "Nothing will get fixed until the Head Office looks at it, but I think they'll be pleasantly surprised that only a couple of devices need fixing, and not dozens or more. Not my problem or role these days, although I did foolishly volunteer to help."

Leo: Good man.

Steve: And I saw the following on my Amazon app on my iPhone. I got a little popup that said: "A small number of deliveries may arrive a day later than anticipated due to a third-party technology outage."

Leo: I saw that too, yeah.

Steve: Yup. Meanwhile in the U.S., almost all airlines were grounded with all their flights cancelled. One, however, was apparently flying the friendly skies all by itself for the day.

Leo: Before you repeat this story, it's been debunked.

Steve: I'm not surprised.

Leo: Yeah. It didn't seem possible.

Steve: Yes. Digital Trends reported under the headline "A Windows version from 1992 is saving Southwest's butt right now." Anyway, yeah.

Leo: Southwest got saved because they didn't use CrowdStrike is how they got saved, not because they were using Windows 3.1.

Steve: Yes, exactly. There are companies all over the world who are not CrowdStrike users.

Leo: Exactly.

Steve: Exactly. And so it was only those who had this csagent.sys device driver loading at boot time in their kernel that had this problem.

Leo: Yup, yup.

Steve: So, and I think that this was made more fun of because in the past, remember that Southwest Airlines has come under fire for having outdated systems.

Leo: Right, yes. But not that outdated.

Steve: Yeah, they had scheduling systems they hadn't updated for a long time.

Leo: Yeah. It's an interesting story, and somebody on Mastodon kind of went through it. And really this happens a lot in journalism nowadays. Somebody tweeted that the Southwest scheduling software looked like it was Windows 95. It wasn't, but looked that way. It got picked up and, like Telephone, it got elaborated to the point where DigiTrends and a number of other outlets, including I might add myself, reported this story. And then we found out it was, you know, Southwest that confirmed it.

Steve: Yeah. And really, I mean, even I'm having problems today on Windows 7 because, you know...

Leo: Yeah, you can't run Windows 3.1.

Steve: ...increasingly things are saying, what are you thinking, Gibson? Like, what is wrong with you? So, yeah.

Leo: You'd have to really, really work hard to keep 3.1 up and running, I think.

Steve: Yeah. So I was initially going to share a bunch of TechCrunch's coverage, but then yesterday Catalin Cimpanu, the editor of the Risky Business newsletter, produced such a perfect summary of this event that only one important point that TechCrunch raised made it - later - into today's podcast, which I'll get to in a minute. But first, here's Catalin's summary, which was just - it's perfect.

So he writes: "Around 8.5 million Windows systems went down on Friday in one of the worst IT outages in history. The incident was caused by a faulty configuration update to the CrowdStrike Falcon security software that caused Windows computers to crash with a Blue Screen of Death." Paul, we realize that's not what it is, thank you. "Since CrowdStrike Falcon is an enterprise-centric EDR, the incident caused crucial IT systems to go down in all the places you don't usually want them to go down. Outages were reported in places like airports, hospitals, banks, energy grids, news organizations, and loads of official government agencies.

"Planes were grounded across several countries, 911 emergency systems went down, hospitals canceled medical procedures, ATMs went offline, stock trading stopped, buses and trains were delayed, ships got stuck in ports, border and customs checks stopped, Windows-based online services went down." He says, for example, ICANN. "And there's even an unconfirmed report that one nuclear facility was affected. The Mercedes F1 team, where CrowdStrike is a main sponsor, had to deal with the aftermath, hindering engineers from preparing the cars for the upcoming Hungarian GP. Heck," he wrote, "even Russia had to deal with some outages." Whoops. I guess they're not quite Windows-free yet over there.

He says: "It was a cluster you-know-what on so many levels that it is hard to put into words how much of the world was upended on Friday, with some outages extending into the weekend. Reddit is full of horror stories where admins lost their jobs, faced legal threats, or were forced to sleep at their workplace to help restore networks. There are reports of companies having tens of thousands of systems affected by the update.

"The recovery steps aren't a walk in the park, either. It's not like CrowdStrike or Microsoft could have shipped a new update and fixed things in the span of a few minutes. Instead, users had to boot Windows into Safe Mode and search and delete a very specific file. The recovery cannot be fully or remotely automated, and an operator must go through the process on each affected system. Microsoft has also released a recovery tool which creates a bootable USB drive that IT admins can use to more quickly recover impacted machines, but an operator still needs to be in front of an affected device.

"For some super lucky users, the BSOD error corrected itself just by constantly rebooting affected systems. Apparently, some systems were able to gain short enough access to networking capabilities to download the fixed CrowdStrike update file and overwrite the old buggy one. However, this is not the universal recommended fix. There are people reporting that they managed to fix their systems after three reboots, while others needed tens of reboots. It took hours for the debug information to make its way downstream, meaning some of the world's largest companies had to bring their businesses to a halt,

losing probably billions in the process." And he said: "Extremely rough estimation, but probably in the correct range."

He writes: "Unfortunately, the Internet is also full of idiots willing to share their dumb opinions. In the Year of the Lord 2024, we had people argue that it's time to ditch security products since they can cause this type of outage. Oh, yes, that's the solution (eye-roll). But CrowdStrike's blunder is not unique, or new for that matter. Something similar impacted loads of other vendors before, from Panda Security to Kaspersky and McAfee. Ironically, CrowdStrike's founder and CEO George Kurtz was McAfee's CTO at the time, but don't go spinning conspiracy theories about it. It doesn't actually mean that much."

He writes: "Stuff like this tends to happen, and quite a lot. As an infosec reporter, I stopped covering these antivirus update blunders after my first or second year because there were so many, and the articles were just repetitive. Most impact only a small subset of users, typically on a particular platform or hardware specification. They usually have the same devastating impact, causing BSOD errors and crashing systems because of the nature of security software itself, which needs to run inside the operating system kernel so it can tap into everything that happens on a PC.

"CrowdStrike released an initial post-mortem report of the faulty update on Saturday. It blamed the issue on what the company calls a 'channel file' update, which are special files that update the Falcon endpoint detection and response" - that's EDR, that's what EDR stands for - "client with new techniques abused by threat actors. In this case, it was Channel File 291." And then he gives us the full file name C-00000291*.sys that causes the crashes.

"CrowdStrike says this file is supposed to update the Falcon EDR to detect malware that abuses Windows named pipes to communicate with its command and control server. Such techniques were recently added to several C2 frameworks - tools used by threat actors and penetration testing teams - and CrowdStrike wanted to be on top of the new technique. The company says the Falcon update file, unfortunately" - yeah, unfortunately - "triggered a logic error. Since Falcon ran in the Windows kernel, the error brought down the house and caused Windows to crash with a BSOD. After that, it was just a house of cards. As the update was delivered to more and more CrowdStrike customers, the dominos started falling all over the world.

"Kurtz [the CEO] was adamant on Friday that this was just an error on the company's part and made it explicitly clear that there was no cyberattack against its systems. U.S. government officials also echoed the same thing. For now, CrowdStrike seems to be focused on bringing its customers back online. The incident is likely to have some major repercussions going beyond the actual technical details and the global outages. What they will be, I cannot be sure," he writes, "but I smell some politicians waiting to pounce on it with some 'ideas,'" he has in quotes. Oh, great.

"This might also be the perfect opportunity/excuse for Microsoft to go with Apple's route and kick most security vendors and drivers out of the kernel. But before that, Microsoft might need to convince the EU to dismiss a 2009 agreement first. Per this agreement, Microsoft cannot wall off its OS from third-party security tools. The EU and Microsoft reached this arrangement following an anti-competitive complaint filed by security software vendors after Microsoft entered the cybersecurity and AV market with Defender, with vendors fearing Microsoft would use its control over Windows to put everyone out of business by neutering their products." Doesn't this sound like, no, sorry, we can't cancel third-party cookies because some people are making money with them. Right.

He writes: "After the recent Chinese and Russian hacks of Microsoft cloud infrastructure, we now know very well what happens when Microsoft has a dominant market position,

and it's never a good thing, so the existence of this agreement isn't such a bad idea. If Microsoft wants to kick security software out of the kernel, Defender needs to lose it, too. Unfortunately, blinding security tools to the kernel now puts everyone in the iOS quandary, where everyone loses visibility into what happens on a system. That's not such a good idea, either. So we're back with this argument where we started.

"In closing, just be aware that threat actors are registering hundreds of CrowdStrike-related domains that will most likely be used in spear-phishing and malware delivery campaigns."

Leo: They're so evil. God.

Steve: "It's honestly one of the best and easiest phishing opportunities we've had in a while."

So as suggested by this week's Picture of the Week, which shows the Windows kernel crash dump resulting from CrowdStrike's detection file update, I will be getting down to the nitty-gritty details that underlie exactly what happened. But I want to first finish laying out the entire story. The part of TechCrunch's coverage that I wanted to include was their writing this: "CrowdStrike, founded in 2011, has quickly grown into a cybersecurity giant. Today, the company provides software and services to 29,000 corporate customers, including around half of Fortune 500 companies, 43 out of 50 U.S. states, and eight out of the top 10 tech firms, according to its website. The company's cybersecurity software, Falcon, is used by enterprises to manage security on millions of computers around the world." And now we know exactly how many millions. "These businesses include large corporations, hospitals, transportation hubs, and government departments. Most consumer devices do not run Falcon and are unaffected by this outage."

And with that lead-up, this was the point: "One of the company's biggest recent claims to fame was when it caught a group of Russian government hackers breaking into the Democratic National Committee ahead of the 2016 U.S. presidential election. CrowdStrike is also known for using memorable animal-themed names for the hacking groups it tracks based on their nationality, such as Fancy Bear..."

Leo: Oh, that's where that came from. Ah.

Steve: "...believed to be part of Russia's General Staff Main Intelligence Directorate, or GRU; Cozy Bear, believed to be part of Russia's Foreign Intelligence Service, or SVR; Gothic Panda, believed to be a Chinese government group; and Charming Kitten, believed to be an Iranian state-backed group. The company even makes action figures to represent these groups, which it sells as swag."

Leo: I want them. Oh, cool.

Steve: "CrowdStrike is so big it's one of the sponsors of the Mercedes F1 team, and this year even aired a Super Bowl ad a first for a cybersecurity company."

Leo: And they were also one of the first cybersecurity companies to advertise on this show, Steve.

Steve: Right.

Leo: In fact, we interviewed the CTO some time ago. It's an impressive company. You know, I'm very curious to hear what happened here.

Steve: So as I have here written, I have not counted the number of times this podcast has mentioned CrowdStrike. It's certainly been so many times that their name will be quite familiar to everyone who's been listening for more than a short while. And not one of those previous mentions was due to some horrible catastrophe they caused. No. As the writer for TechCrunch reminds us, CrowdStrike has been quite instrumental in detecting, tracking, and uncovering some of today's most recent and pernicious malware campaigns and the threat actor groups behind them.

How are they able to do this? It's entirely due to exactly this same Falcon Sensor instrumentation system that's been spread far and wide around the world. It's this sensor network that gives them the visibility into what those 8.5 million machines that have been running it are encountering day to day in the field.

Leo: Exactly. Exactly.

Steve: And we need that visibility.

Leo: By the way, here is the Aquatic Panda figurine, currently $28 on the CrowdStrike swag shop.

Steve: Wow, you've got to be really deep into whatever. Wow.

Leo: I want Fancy Bear. Okay. Sorry.

Steve: Okay. So this is not to in any way excuse the inexcusable. Make no mistake that what they just caused to happen was inexcusable. But any coherent answer to the question "What is CrowdStrike and why in the world are we putting up with them?" should in fairness acknowledge that the same network that just crippled much of the world's cyber operations has also been responsible for discovering malicious activities and protecting not only their own customers, but all of the rest of us, as well.

Catalin referred to George Kurtz, late of McAfee and now CrowdStrike's Founder and CEO. George's note about this was addressed to "Valued Customers and Partners." He wrote: "I want to sincerely apologize directly to all of you for today's outage. All of CrowdStrike understands the gravity and impact of the situation. We quickly identified the issue and deployed a fix, allowing us to focus diligently on restoring customer systems as our highest priority. The outage was caused by a defect found in a Falcon content update for Windows hosts. Mac and Linux hosts are not impacted. This was not a cyberattack. We're working closely with impacted customers and partners to ensure that all systems are restored, so you can deliver the services your customers rely on.

"CrowdStrike is operating normally, and this issue does not affect our Falcon platform systems. There's no impact to any protection if the Falcon sensor is installed. Falcon

Complete and Falcon OverWatch services are not disrupted. We will provide continuous updates through our Support Portal via the CrowdStrike blog. Please continue to visit these sites for the latest updates. We've mobilized all of CrowdStrike to help you and your teams. If you have questions or need additional support, please reach out to your CrowdStrike representative or Technical Support.

"We know that adversaries and bad actors will try to exploit events like this. I encourage everyone to remain vigilant and ensure that you're engaging with official CrowdStrike representatives. Our blog and technical support will continue to be the official channels for the latest updates. Nothing is more important to me than the trust and confidence that our customers and partners have put into CrowdStrike. As we resolve this incident, you have my commitment to provide full transparency on how this occurred and steps we're taking to prevent anything like this from happening again. George Kurtz, CrowdStrike Founder and CEO."

And with that, this podcast can now dig into some of the interesting and more technical nitty-gritty that will answer all of the remaining questions. Leo, let's take our final break, and then we're going to do that.

Leo: All right. Fascinating stuff. I wonder if CrowdStrike ever will really tell us what happened in greater detail than they have.

Steve: The industry has reverse engineered, and I'll be talking about that.

Leo: Oh, good. Okay, well, that's one way to find out. Good. All right. We'll get to that in a moment. All right. So a little reverse engineering, and maybe we can figure out what happened here. You know, the email you read a few emails ago is the question everybody's asking. How could a company do this? How could they not have tested? How could they not know? But to understand that, we need to understand what happened.

Steve: Right.

Leo: And that's your job.

Steve: Okay. So let's begin with CrowdStrike's predictably not-very-technical update which they titled: "Falcon update for Windows hosts technical details." And I'm doing this because I need to establish the context for what we're going to talk about next. So they ask themselves the question, what happened? And they answer: "On July 19th, 2024 at 04:09 UTC, as part of ongoing operations, CrowdStrike released a sensor configuration update to Windows systems. Sensor configuration updates are an ongoing part of the protection mechanisms of the Falcon platform. This configuration update triggered a logic error resulting in a system crash and blue screen (BSOD)" - they actually said that - "on impacted systems.

"The sensor configuration update that caused the system crash was remediated on Friday, July 19th, 2024 at 05:27 UTC." So 4:09 it happened; 5:27 they fixed it. Of course the damage was done; right? You're stuck in a boot loop, you can't update yourself. They said: "This issue is not the result of or related to a cyberattack. Customers running Falcon sensor for Windows version 7.11 and above, that were online between Friday at 04:09 UTC and up to 05:27 when they fixed it, may be impacted. Systems running

Falcon sensor for Windows version 7.11 and above that downloaded the updated configuration from, again, those same two times, were susceptible to a system crash." Yeah, no kidding. Find somebody who doesn't know that.

Okay. Then Configuration File Primer, however you want to pronounce it: "The configuration files mentioned above are referred to as Channel Files and are part of the behavioral protection mechanisms used by the Falcon sensor. Updates to Channel Files are a normal part of the sensor's operation and occur several times a day in response to novel tactics, techniques, and procedures" - you know, TTPs - "discovered by CrowdStrike. This is not a new process; the architecture has been in place since Falcon's inception."

Then Technical Details: "On Windows systems, Channel Files reside in the following directory: C:\Windows\System32\drivers\CrowdStrike. So they have their own subdirectory under drivers, and have a filename that starts with C-." Then they said: "Each channel file is assigned a number as a unique identifier. The impacted Channel File in this event is 291 and will have a filename that starts with C-00000291- and ends with a .sys extension. Although Channel Files end with the .sys extension, they are not kernel drivers. Channel File 291 controls how Falcon evaluates named pipe execution" - and I'll talk about that in a second - "on Windows systems.

"Named pipes are used for normal, interprocess, or intersystem communication in Windows. The update that occurred at 04:09 UTC was designed to target newly observed, malicious named pipes being used by common C2 framework" - you know, command-and-control frameworks - "in cyberattacks. The configuration update triggered a logic error that resulted in an operating system crash. CrowdStrike has corrected the logic error by updating the content in Channel File 291. No additional changes to Channel File 291 beyond the updated logic will be deployed." Meaning they're not changing their driver.

"Falcon is still evaluating and protecting against the abuse of named pipes. This is not related to null bytes contained within Channel File 291 or any other Channel File." That was another, you know, specious rumor that took hold of the Internet that the file was all nulls.

"The most up-to-date remediation recommendations and information can be found on our blog or in the Support Portal. We understand that some customers may have specific support needs, and we ask them to contact us directly. Systems that are not currently impacted will continue to operate as expected, continue to provide protection, and have no risk of experiencing this event in the future. Systems running Linux or macOS do not use Channel File 291 and were not impacted."

And they finish with this "Root Cause Analysis." They said: "We understand how this issue occurred, and we are doing a thorough root cause analysis to determine how this logic flaw occurred. This effort will be ongoing. We are committed to identifying any foundational or workflow improvements that we can make to strengthen our process. We will update our findings in the root cause analysis as the investigation progresses."

Okay. So now we have a comprehensive understanding of what happens to today's truly IT-dependent global operational existence if around 1%, or about 8.5 million, extra-specially secured and running Windows OS-based machines are all caused to spontaneously crash and refuse to return to operation. It's not good. I was unable to use my mobile app to remotely order up my morning five shots of espresso from Starbucks. This is unacceptable. Something must be done.

Before we address the most burning question of all, which is how CrowdStrike and their systems could have possibly allowed something so devastating to occur, let's first answer

the question of why Windows' own recovery systems were also unable to be effective in recovery from this. Many years ago this podcast looked at a perfect example of a so-called "rootkit" in the form of Sony Entertainment's deliberately installed DRM, digital rights management software.

We saw how a normal directory listing of files could be "edited" on the fly to remove all visibility of specific files, turning them into ghosts. The files were there, but we could not see them. Sony's programmers did that to hide the presence of their own DRM. That Sony rootkit demonstrated that we could not believe our own eyes, and it perfectly drove home just how much we depend upon the integrity of the underlying operating system to tell us the truth. We take for granted that when we ask for a file listing, it's going to show all of them to us. Why would it lie? Why indeed?

If malware of that sort is able to infect the core operation of an operating system, then it's able to hide itself and strongly protect itself from both discovery and removal. That was Sony's goal with their rootkit. This fact creates a competition between the good guys and the bad guys to get in and establish the first foothold in an operating system because the entity that arrives on the scene first is able to use its placement to defend itself from anything that might happen afterwards.

For that reason, the actual CrowdStrike pseudo-device driver - not the mistaken channel file, but the actual driver that later reads those channel files - which Microsoft themselves previously examined, approved, and digitally signed, was given the highest honor of being flagged as a "boot-start" device. Microsoft's own documentation says: "A boot-start driver is a driver for a device that must be installed to start the Microsoft Windows operating system."

In other words, once device driver code is present containing that "boot-start" flag, Windows is being told and believes that it must load that driver in order to successfully start itself running. For all it knows, that driver is for the mass storage RAID array that contains Windows itself; and if that driver is not installed and running by the time the motherboard firmware has finished loading the various pieces, Windows will be unable to access its own mass storage. So Windows may not know why, but it does know that a device driver that carries a valid Microsoft digital signature that has the "boot-start" designation must be loaded. So it will be loaded and initialized in order for Windows to successfully boot. And this brings us to CrowdStrike's boot-start driver.

CrowdStrike's engineers designed a powerful driver that's designed to significantly augment Windows' own antimalware defenses. When it's loaded into Windows, it hooks many of Windows' Application Programming Interface (API) functions. What this means is that it places itself in front of Windows, so that any code that would normally ask Windows to do something on its behalf will instead be asking CrowdStrike's code. And only if CrowdStrike sees nothing wrong with the request will CrowdStrike, in turn, pass that application's request on to Windows. It creates a defensive wrapper shell around the Windows kernel. And if this sounds exactly like what a rootkit does, you would be right. CrowdStrike's driver kit is in the root. It needs to, to get its job done.

What little we've learned about the specifics of this CrowdStrike failure is that it involved named pipes. And that again makes sense since named pipes are used extensively within Windows for interprocess communications. It's pretty much "the" way clients obtain services.

The on-the-fly code-signing service which I created before the release of SpinRite 6.1 is started by Windows after it finishes booting. When that service starts up, the one I created, it uses the Windows API to create a "named pipe" with a unique name. Then later, when a user purchases a copy of SpinRite, GRC's web server opens a dynamic connection to the code-signing service by opening a named pipe of the same name. This

deliberate name collision connects the two separate events to establish a highly efficient interaction which allows the service and its client to then negotiate the details of what needs to be done.

CrowdStrike told us that malware had been seen using the named pipes API for communication with its command and control servers, so they needed to "hook" Windows named-pipes API in order to monitor the system for, and to possibly block, that activity. And clearly something went terribly wrong.

For reasons that CrowdStrike is still being silent about, the presence of this channel file caused a bad parameter to be passed to a function. This week's Picture of the Week is a snapshot of the crash event that brought down all of those 8.5 million Windows machines. We see a procedure being called with two parameters. The first parameter is zero, and the second parameter has the value of hexadecimal 9C, nine Charlie, which is decimal 156. That value of 156 is loaded into the CPU's r8 register, where it's then used as a pointer to point to the value in question.

So then the CPU is asked to load the 32-bit value from that location into the CPU's r9 register. The only problem is, the way the system's memory is mapped, there's nothing, no memory, located at location 156. The processor, realizing that it has just been asked to load 32 bits of memory that does not exist, figures that something must have gone very, very wrong somewhere, so it panics.

Leo: And Steve doesn't get his venti latte. It's just that simple.

Steve: I can't get my five shots. I have to actually show up in person. It's like the Pilgrims, Leo. Suddenly we're reduced to fighting Indians.

Leo: It's not really funny for the, I mean, some people got stranded for two or three days in the airport.

Steve: Well, and IT admins lost their jobs. There were people fired because they were blamed for this.

Leo: Oh, that's sad. That's really sad.

Steve: Yes.

Leo: No one should be blamed for this.

Steve: Yes.

Leo: Except CrowdStrike, I guess. So...

Steve: Okay. So...

Leo: It's interesting. So it jumped to a kernel panic because it jumped to an area of memory that doesn't exist. That's what a machine is supposed to do. That's what a blue screen does. That's the whole point.

Steve: Right. Yes. The kernel panics, and it declares an emergency.

Leo: It's over, yeah, right.

Steve: Yes. Now, what I just described is a very, exactly as you said, Leo, a very typical series of events which precipitates what the entire world has come to know as the Blue Screen of Death.

Leo: Yeah.

Steve: It's something that should never happen. But hardware is not perfect, and neither are people.

Leo: And I've seen it, it's a green screen on the Mac. It's a black screen...

Steve: A black screen on Linux.

Leo: On Linux, yeah. So every operating system does this. There's no reasonable way to recover from a failure like this.

Steve: No. Things that are not supposed to ever happen, happen anyway, is the point.

Leo: Right.

Steve: And just to be clear, if an application running on top of the Windows operating system as one of its client applications, were to ever ask the processor to do something impossible, like read from non-existent memory or divide by zero, Windows could and would just shrug it off and allow the app to crash. Windows might display some mumbo-jumbo on the screen that would be entirely meaningless to the app's user, but life would go on without any other app in the system caring or being the wiser. The crucial difference is where this unrecoverable error occurs. When it occurs deep within the operating system kernel itself, there's no one to call. It's game over. And the thing that gets terminated is the operating system itself.

Now, we do not know exactly where that aberrant value of 156 came from. There's been speculation that it looks like an offset into a structure, and that would make sense. So that if the structure's pointer itself were null, an offset like this might result, and then be fed to a function that was expecting to be pointing to a member of a valid object, instead of out into hyperspace. Some have observed that CrowdStrike's channel file contains nulls.

But the unsatisfying reality is that all we have today is conjecture. There is no question that CrowdStrike definitely and absolutely already knows exactly what happened, but they're not saying. I have absolutely zero doubt that their corporate attorneys clamped down the cone of silence over CrowdStrike so rapidly and forcibly that no one inside CrowdStrike even dares to mumble about this in their sleep. I mean, this is, you know, to say that there will be repercussions has got to be the, you know, just doesn't even begin.

Okay. There is very likely good news on that front, however, that is, what actually happened. CrowdStrike's driver and its associated Channel File 291, now infamous, which together triggered this catastrophe, still exist in the world. And the world contains a great many curious engineers who are gifted at reverse engineering exactly such disasters. So I doubt we'll be waiting long before we receive a beautifully detailed explanation of exactly what happened, at least at the client end. But it will not be forthcoming from CrowdStrike, and it won't need to be because we're going to figure this out as an industry.

But now we face the final biggest question of all, which is how could CrowdStrike, or any other company for that matter, having as much at stake and to lose from being the proximate cause of this incredibly expensive earthshaking catastrophe, possibly have allowed this to happen? The thing that everyone wants to know is how could CrowdStrike have possibly allowed this defective Channel 291 File to ever be deployed at all? And when I say that everyone wants to know, that now of course includes members of the United States Congress who are demanding that George Kurtz present himself before them on his knees with his head bowed and his neck exposed.

I have a theory. Not long ago, just a few months ago, Microsoft asked us to believe that an incredibly unlikely, credulity-stretching chain of events - as I recall it was five unlikely failures, each of which were required for a private key to leak - enabled a malicious Chinese actor to obtain Microsoft's private key, which led to that large and embarrassing Outlook 365 email compromise. Assuming that all of this was true, it's clear that so-called "black swan" events can occur, no matter how unlikely they may be. So one view of this is that CrowdStrike does indeed have multiple redundant systems in place to guard against exactly such an occurrence. After all, how could they not? Yet, as we're told happened to Microsoft, despite all of that, something was still able to go horribly wrong.

But I said "one view" above because there's another possibility. It's not outside the box to imagine that this particular failure path, whatever it was, was actually never expected or believed to be possible by the system's designers, so there may not actually have been any pre-release sandboxed sanity testing being done to those multiple times per day malware pattern updates that CrowdStrike was publishing to the world. And remember that CrowdStrike had been doing this successfully and without any massive incident such as this for many, many years.

Now, I understand that to someone who does not code, like those in the U.S. Congress, this is going to sound quite nutty. But developers typically only test for things that they know or imagine might possibly go wrong. If a coder adds two positive numbers, they don't check to make sure that the result is greater than zero because it must be. So I would not be surprised to learn that what happened slipped past CrowdStrike's own sanity-checking verifiers because it was something that was not believed to be possible. And while now, yes, a final last-stage sanity check against live systems prior to deployment seems like the obviously important and necessary addition, either it must not have seemed necessary and hasn't ever been in the past, or it IS present, and nevertheless it somehow failed, much as those five sequential and separate failures within Microsoft allowed their closely held secret to escape.

Also, we don't know exactly what sort of CDN (content delivery network) they're using. If 8.5 million instances of their device driver are reaching out for more or less continual updates 24/7, they're likely using some distribution network to spread those updates. What if everything WAS good at CrowdStrike's end, but the upload of that file to the cloud somehow "glitched" during transit, allowing that glitched file to then be distributed to the world's Windows machines? Of course, we have means for protecting against this, too, digital signing by the authorized sender and signature verification by the recipient. We would certainly hope that these updates are signed, and any failure of signature verification would prevent the file's use. But we've seen that not happening, too, in the past. So distribution failure should, you know, we would hope it does not explain the trouble, but it might.

I'll finish with the acknowledgement that what has happened is so bad that I doubt we're going to soon learn the truth from CrowdStrike, who wrote: "We understand how this issue occurred, and we are doing a thorough root cause analysis to determine how this logic flaw occurred. This effort will be ongoing." Okay. That's just gibberish. You say you understand how this issue occurred because you don't want to seem totally incompetent, but you're not sharing that understanding because you're too busy determining how it occurred. Right.

Again, I expect that our industry's beloved reverse engineers will have a great deal to share long before CrowdStrike's attorneys finish proofreading and massaging the carefully worded testimony which George Kurtz will soon be delivering in front of Congress.

Leo: So really any number of things could have happened. It's hard to ascribe blame if you don't know, you know, like they may have done testing, and it could have been a flaw in the CDN. So, yeah. We just need to hear more from them. Probably won't. Probably won't.

Steve: Yeah. I've heard reports, again unconfirmed, that the files are not signed. If the files are not signed, that's a crime.

Leo: Yeah.

Steve: Because if that's true, it would mean that it would be possible for a glitch, like a transmission glitch in the content delivery system to cause this.

Leo: Yeah. Or a miscreant in the middle. Just stick another one in.

Steve: A miscreant in the middle.

Leo: Yeah.

Steve: But, you know, really we can't rule that out.

Leo: And I do blame Microsoft a little bit. I guess they blame the EU. But to allow an interpreter to run in ring zero, and then to allow these unsigned files, if they're

unsigned, to run in the interpreter is a very risky proposition. Right? You've talked all along about how interpreters are the number one cause for these problems.

Steve: Yes. And the other thing, too, is again, if I didn't have a day job, I would be looking around for a copy of this file. The burning question to me is whether there's executable code in this. It may not itself be an executable file. But it could have...

Leo: But Falcon is an interpreter, I think.

Steve: Okay. So then it contains p-code.

Leo: Right.

Steve: And the p-code could have been bad, and that caused a problem in the interpreter.

Leo: Right, right.

Steve: So, yeah.

Leo: So, you know, these systems are based - it's not quite an antivirus signature, but it's basically the same idea, which is they are...

Steve: Well, it has to do with behavioral analysis.

Leo: Exactly.

Steve: Behavioral analysis requires more of an algorithmic approach.

Leo: It's heuristic, yes. So it has to be more of a program than let's look for a string.

Steve: Right.

Leo: And so I think that's the thesis is that these little sensors are p-code, and then this way basically you have a security system running in ring zero that can be modified on the fly.

Steve: Yes.

Leo: Problem. Right there.

Steve: Dangerous. Dangerous.

Leo: Problem. Dangerous. And you might, you know, the Falcon can be signed. Probably, I'm sure the Falcon would sign it, wouldn't run in ring zero.

Steve: Microsoft signed the Falcon driver.

Leo: Yeah, of course.

Steve: You can't run...

Leo: It can't run in ring zero.

Steve: You can't run in the kernel unless it's blessed by Microsoft.

Leo: And one of the things we talked about on MacBreak Weekly is that Apple has very carefully, but fairly aggressively, removed the ability to put kernel extensions into macOS for this very reason. Right? They're not required to by the EU. You know, nobody's saying Apple's not big enough for anybody to say, well, you've got to allow these security guys to put stuff in the kernel. There are ways around it. You can run kernel extensions. But it takes - you have to jump through a lot of hoops, and Apple basically disallows it. And they've been slowly moving that direction.

Steve: It might be that the upshot of this is another level of disaster recovery, which Microsoft has not yet implemented.

Leo: Maybe that's it, yeah.

Steve: Because, you know, it did - the fact that it turned the screen blue and put up some text means that something [crosstalk].

Leo: Something's running, yeah.

Steve: Yes. So, yeah, I mean, and what happens is an exception is created when you try to read from nonexistent memory. That exception goes to code that is an exception handler. And what it could do is set something that then causes a reboot into a different copy of the OS, for example. I mean, they could, you know, or as we know, you could reboot under a manually forced Safe Mode in order to keep that Falcon driver from running.

Leo: Precisely. And I think a number of people recommended that, that this would be an, not easy, but this would be an implementable solution, which is if something caused a crash, you don't load it on the next boot and see what happens then. I can see problems with that also. But I think that something will have to be done.

Steve: Well, yes. For example, if it was the system's RAID array driver...

Leo: Right, that's not going to work, obviously.

Steve: Then, yes.

Leo: Yeah. But if it's a third-party signed driver, I mean, any driver can do this, by the way. Drivers generally do run in ring zero because that's how you talk to the hardware.

Steve: Right.

Leo: But remember Microsoft created HAL, the Hardware Abstraction Layer, for that very reason, so that you could talk to a variety of hardware, and you didn't have to operate at such a low level. Their own interpreter, in effect. You know, it's a very - seems solvable, but obviously I don't have enough information to know what really happened.

Steve: Well, no one does.

Leo: No one does.

Steve: They're not saying, and they're not going to say.

Leo: Yeah.

Steve: I mean, their entire existence is hanging by a thread at the moment.

Leo: Oh, yeah, yeah.

Steve: I mean, this was a big deal. And they're going to have to explain, not only to Congress, but to the world, exactly what this root cause nonsense is.

Leo: Well, and that's one of the reasons you and I have both for a long time said end users should not be running antivirus software because, as a necessity, it runs at a low level and has permissions that normal software doesn't. And we've seen it again and again that that can become a vector for malware.

Steve: Yup.

Leo: There's no evidence that this was intentional or malware, though, at this point.

Steve: None.

Leo: None.

Steve: There's no evidence either way. You know, they of course don't want it to be a cyberattack. They don't want it to be deliberate. But they haven't yet explained how a bad file got out to 8.5 million machines.

Leo: They're in a bad position because it's either stupidity, incompetence, or a bad guy. I mean, it's hard. That's why they haven't said anything; right? Because if they tell us what really happened, it's very possible people will say, well, that was dumb.

Steve: Yeah.

Leo: I think if there were a good excuse, they would have said what it was already.

Steve: Right.

Leo: Right?

Steve: Right.

Leo: Oh, it got corrupted in the CDN.

Steve: This will not be the last time we are talking about CrowdStrike.

Leo: Yeah. You know, and as you pointed out early on, they were the good guys. They are the good guys. They do good work.

Steve: Oh, my god, they have - the power of that sensor network.

Leo: It's incredible.

Steve: The way we are finding bad guys on a global scale. They are absolutely, you know, we need them there. They just tripped up.

Leo: Right. I talked to the CTO, and that's exactly what we talked about is their sensor network and how many billions of signals they got. You know, some huge number every minute.

Steve: It's astonishing.

Leo: Yeah. It was so valuable because they could see threats in real time, as they happened, catch zero-days right then. Ah. This is fascinating. Thank you. As always, Steve does a great job explaining this. And I was waiting all weekend to get here and find out what Steve had to say.