Security Now! #984 - 07-23-24

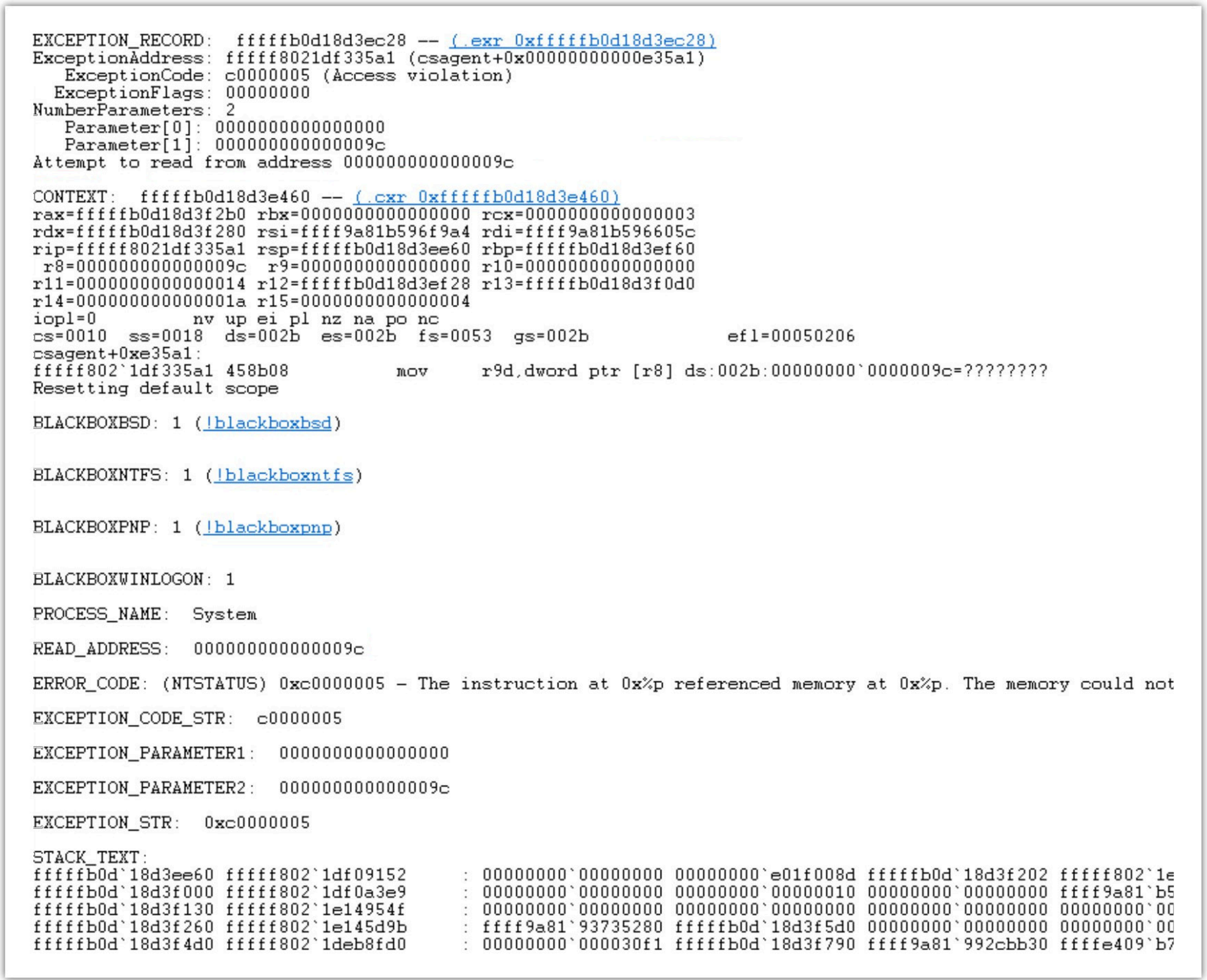
CrowdStruck

“The Legend of Channel File 291”

This week on Security Now!

What do we know about how the FBI broke into the smartphone of Trump's deceased would-be assassin? Cisco scored another of the very rare CVSS 10.0s for a serious remote authentication vulnerability. If you're affected you must update! Untrusted Entrust's plan for the future has been revealed. Surprisingly, Google loses the anti-3rd-party cookie battle. 3rd-party cookies stay perhaps forever. I share a few more interesting experiences from GRC's weekly Security Now podcast mailings. We now know where the seemingly “flaky” name “Snowflake” came from. And after sharing a collection of interesting listener feedback follow-ups from recent discussions, we 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.

The simple memory pointer mistake that stalled the world:



```
EXCEPTION_RECORD:  fffffb0d18d3ec28 -- (.exr 0xfffffb0d18d3ec28)
ExceptionAddress: fffff8021df335a1 (csagent+0x00000000000e35a1)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: 000000000000009c
Attempt to read from address 000000000000009c

CONTEXT:  fffffb0d18d3e460 -- (.cxr 0xfffffb0d18d3e460)
rax=fffffb0d18d3f2b0 rbx=0000000000000000 rcx=0000000000000003
rdx=fffffb0d18d3f280 rsi=ffff9a81b596f9a4 rdi=ffff9a81b596605c
rip=fffff8021df335a1 rsp=fffffb0d18d3ee60 rbp=fffffb0d18d3ef60
 r8=000000000000009c  r9=0000000000000000 r10=0000000000000000
r11=0000000000000014 r12=fffffb0d18d3ef28 r13=fffffb0d18d3f0d0
r14=000000000000001a r15=0000000000000004
iopl=0         nv up ei pl nz na po nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00050206
csagent+0xe35a1:
fffff802`1df335a1 458b08          mov     r9d,dword ptr [r8] ds:002b:00000000`0000009c=????????
Resetting default scope

BLACKBOXBSD: 1 (!blackboxbsd)


BLACKBOXNTFS: 1 (!blackboxntfs)


BLACKBOXPNP: 1 (!blackboxpnp)


BLACKBOXWINLOGON: 1

PROCESS_NAME:  System

READ_ADDRESS:  000000000000009c

ERROR_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%p referenced memory at 0x%p. The memory could not

EXCEPTION_CODE_STR:  c0000005

EXCEPTION_PARAMETER1:  0000000000000000

EXCEPTION_PARAMETER2:  000000000000009c

EXCEPTION_STR:  0xc0000005

STACK_TEXT:
fffffb0d`18d3ee60 fffff802`1df09152     : 00000000`00000000 00000000`e01f008d fffffb0d`18d3f202 fffff802`1e
fffffb0d`18d3f000 fffff802`1df0a3e9     : 00000000`00000000 00000000`00000010 00000000`00000000 ffff9a81`b5
fffffb0d`18d3f130 fffff802`1e14954f     : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00
fffffb0d`18d3f260 fffff802`1e145d9b     : ffff9a81`93735280 fffffb0d`18d3f5d0 00000000`00000000 00000000`00
fffffb0d`18d3f4d0 fffff802`1deb8fd0     : 00000000`000030f1 fffffb0d`18d3f790 ffff9a81`992cbb30 ffffe409`b7
```

With the glare of last week's breathtaking world-spanning CrowdStrike-driven Windows OS failure still looming over everything, it's understandably difficult to focus upon anything else. And we will certainly be giving last Friday's event our full attention. But there was some other important news that emerged during the past week which should not be overshadowed.

Security News

Cellebrite unlocks Trump's would-be assassin's phone.

In the wake of the failed attempted assassination of Donald Trump during his recent campaign rally, the FBI has been attempting to learn all it can about the immediately deceased would-be assassin. 20 year-old Thomas Matthew Crooks was using an Android phone that was password locked. We've certainly been here before, haven't we?

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 US federal agencies. 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.

Although the local FBI bureau in Pittsburgh already had a current license for Cellebrite's smartphone cracking software, it was ineffective on Thomas Crooks' newer Samsung device. Undaunted, the FBI reached out to Cellebrite's nearby federal team, which is there in order to collaborate with law enforcement and government agencies.

Within hours, Cellebrite had provided the FBI with the additional support they needed, including some newer software that was not yet released – and 40 minutes later the FBI had Thomas' Samsung Android smartphone unlocked and open for a detailed inspection of the shooter's history.

In other coincident 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"*, writing:

> *Leaked documents reveal that Cellebrite cannot unlock iPhones running iOS 17.4 and later, at least as of the date of publication (April 2024). 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 aim to discover and fix. Others also work to defeat the phone-cracking kit, with (mostly) secure messaging app Signal scoring a big win in 2021, when it managed to booby-trap iPhones to render the kit useless.*
>
> *Back in 2022, 9to5Mac managed to obtain user documentation revealing which iPhone models*

> *the kit could and couldn’t unlock. 404 Media has now done the same with a later document, dated April 2024. As of that date, Cellebrite had not managed to crack iPhones running iOS 17.4 or later, which will be a very large percentage of iPhones.*
>
> *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 iPhone 11 mean those are exceptions. The company 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” to other models. 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. 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 they were obtained. That’s because the cold-boot process blocks the exploit 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 to Android 14.*

The Verge summarized the state of play by writing: *“Phone hacking companies are overstating their capabilities.”* and 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. 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 impenetrable. (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!)

Cisco reported on a CVSS of 10.0

As we know, CVE’s having a CVSS score of 10.0 out of a possible 10 are vanishingly and blessedly rare. Unfortunately, last Wednesday, Cisco reported and acknowledged one of their own. ArsTechnica wrote:

> *On Wednesday, Cisco disclosed 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.*
>
> *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. It’s used by customers*

> *who can’t or don’t want to manage licenses in the cloud, as is more common.*
>
> *In a bulletin, Cisco warns that the product contains a vulnerability that allows hackers to change any account's password. The severity of the vulnerability, tracked as CVE-2024-20419, is rated 10, the maximum score.*
>
> *The Cisco bulletin stated: “This vulnerability is due to improper implementation of the password-change process. An attacker could exploit this vulnerability by sending crafted HTTP requests 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.*
>
> *It’s unclear precisely what an attacker can do after gaining administrative control over the device. One possibility is that the web user interface and application programming interface the attacker gains administrative control over make it possible to pivot to other Cisco devices connected to the same network and, from there, steal data, encrypt files, or perform similar actions. Cisco representatives didn’t immediately respond to an email. This post will be updated if a response comes later.*
>
> *A security update linked to the bulletin fixes the vulnerability. Cisco said it isn’t aware of any evidence that the vulnerability is being actively exploited.*

I’m relaying this, first because it’s one of the ultra-rare 10.0’s, and also because if by any chance any listener of this podcast has authority over this Cisco gear they’ll want to be very sure this has been updated.

Entrust drops the other shoe

As we know, several weeks ago, after a great deal of hand wringing 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. This wasn’t done, as had been done in the past, due to any horrific certificate mis-issuance event, but rather to send a very strong message 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 mis-issued certificates in place and thus saving face with their customers, while other Ca’s 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. Google’s move was, and was meant to be, a very clear demonstration that this game would not tolerate any multiple-year endemic cheating.

The week after we covered this historic event, Todd Wilkinson, Entrust’s President & CEO, formally apologized and said (once again after the industry had lost count of the number of times Entrust had said this before) that they were really truly and seriously going to do better this time. And I suspect that this time they probably will. But that left us with the question: What would Entrust do in the meantime? We’re back here today because Todd is present the answer that they question. He signed the following update:

> *To our TLS customers:*
>
> *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, Google has said that Chrome will no longer accept Entrust public TLS certificates issued after October 31, 2024. Entrust TLS certificates issued prior to October 31 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 have 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 31, 2024, you can continue to request public certificates and receive certificate services directly from Entrust. Here is how this will work:*
>
> - *Continue to order certificates as you have been, under the same pricing model and SLAs.*
> - *Rely on Entrust for certificate lifecycle management, verification, support, and professional services, as we plan to serve as the Registration Authority (RA) for these certificates.*
> - *We will deliver public TLS certificates issued by a CA partner that meets the requirements of the CA/Browser Forum and Entrust.*
>
> *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.*
>
> *To build resilience into your organization, we recommend that you take inventory and renew your Entrust certificates prior to October 31, 2024. These certificates will be trusted through their expiration date, up to 398 days. You can renew your certificates through your certificate lifecycle management solution, automation tool, or the Entrust Certificate Services Portal.*

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 doesn't 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 certificates. Instead, Entrust found SSL.COM, an even smaller CA 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 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.

It's a win-win for the time being, but this does also feel like a temporary backstop solution. It feels as though Entrust does 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 revenue, they will likely be able to retain the customer relationships they have built and will someday be able to again issue certificates under their own name.

Google gives up on removing 3rd-party cookies

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 – which it would subsequently return, thus identifying itself to the website for all subsequent activities. This simple solution enabled the concept of being “logged onto” a website in the same way that users were previously able to login to other online services.

As this podcast’s long time listeners know, I’ve always been very annoyed by the abuse of this simple cookie technology by 3rd-parties, since cookies were purely and expressly intended to be used as a means for maintaining logged-on session state. But the advent of 3rd-party advertisers whose ads poked out onto tens of thousands of websites meant that their 3rd-party cookies could be used to track and profile users as they moved across the web.

For this reason, I’ve been excited and hopeful about Google’s repeated attempts to design a workable alternative, after which they would, they have said, completely eliminate all support for 3rd party cookies. After which the web would, in my opinion, finally be operating the way Mozilla intended without cookie abuse while still offering advertisers what they wanted. The only problem was, the achievement of this goal would also collapse the entire Internet tracking, profiling and data aggregating industry. And for this 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”*. Understanding what Anthony is really saying requires a great deal of reading between the lines. Here’s what he wrote:

> *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.*
>
> *Throughout this process, we’ve received feedback from a wide variety of stakeholders, including regulators like the UK’s Competition and Markets Authority (CMA) 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. 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.*
>
> *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.*

So, 3rd party cookies will remain in Chrome and it appears unlikely that technologies such as 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 a user-interface and whether websites begin insisting that their users enable whatever this is going to be called if they want access to the site's content – oh! And while you're at it, what's a good email address for you?

Miscellany

I thought 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. I found three causes for the trouble caused by last Tuesday's mailing:

First, I verified that the picture-of-the-week's thumbnail image was indeed triggering a false positive detection from some A/V 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 A/V rejection false positive, and this time only 3 of them bounced.

Secondly, after carefully examining the feedback from some of the A/V 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! Got me on that one! I don't know whether enclosing the domain separating dot in square brackets would have rendered that domain name safe to include in email, but I'll make a point of doing that in the future so we'll eventually find out.

The third and final discovery was a complaint about invisible text in the email. I had taken an innocuous looking line from several suggestions about composing email for viewing on a wider range of devices. It was an HTML <div> line with its "display" style set to none and a font point size of 0. It did not contain any text, but that didn't appear to matter. So that's been removed from all future mailings.

So I expect that today's weekly Security Now! Email will be delivered with far greater success than last week's!

I should also note that it was mostly those 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". Okay. Well I am just getting started here so that was expected and it doesn't make me love Hover any less. But I've confirmed with a listener, Paul Sylvester, whom I exchanged email with, that it's possible to log into your Hover webmail console and add email senders to your known good guys list. He did that for both the "mail-manger" and "securitynow" accounts at GRC.com and all trouble immediately disappeared.

Closing The Loop

Jeff Garretson in Yakima, WA (pronounced YAK-i-ma).

> *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 star schema. Hope that helps. Love the show!*

So what Jeff just shared is the "official" reason for the name. In a much more playful mood than they are today, ten years ago 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:

> *One of the questions that we get the most is, "Why did you decide to name the company Snowflake?" 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 fit us in several ways:*
>
> - *Snowflakes are "born in the cloud" For a data warehouse built from the ground up for the cloud, that's very important.*
> - *We love the snow. Most of our founding team, and even our first investor, love to spend time up in the mountains in winter. They even convinced me to try skiing (and took me on a black run my first day).*
>
> - *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.

A listener requesting anonymity shared his experience following the recent CDK Global dealership outage:

> *Hi Steve, I'm a software developer and I develop an interface between my company's software and CDK dealership. 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 all 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 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 don't double book. 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. Regards.*

John Meuser (rymes with User) 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 is that this does invalidate one of the reasons a web developer might use externally hosted resources. Suppose there is 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 in the latest compatible version, they will always have the latest bug fixes. There will always be a difficult balance to be struck for convenience and security.*

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 by specifying the resource's exact version number. But as John notes, that also means that the webpages 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.

Simon (an Aussie in the UK)

> *I suggest MITM now stands for Miscreant In The Middle.*

I like that one a lot. We don’t need to change the abbreviation from MITM, just that the first ‘M’ stands for Miscreant.

Ryan Frederick

> *You said on this week's show that you will make a copilot plus blocker app in assembly, if Microsoft ever releases it. At this time, the only copilot plus 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.*

Ryan makes a very good point. However, we know that there's no way ARM-based Windows machines won't also be able to emulate Intel x86 family instructions to run all Intel-based apps. 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 significant.

Kris Quinby in Riverdale

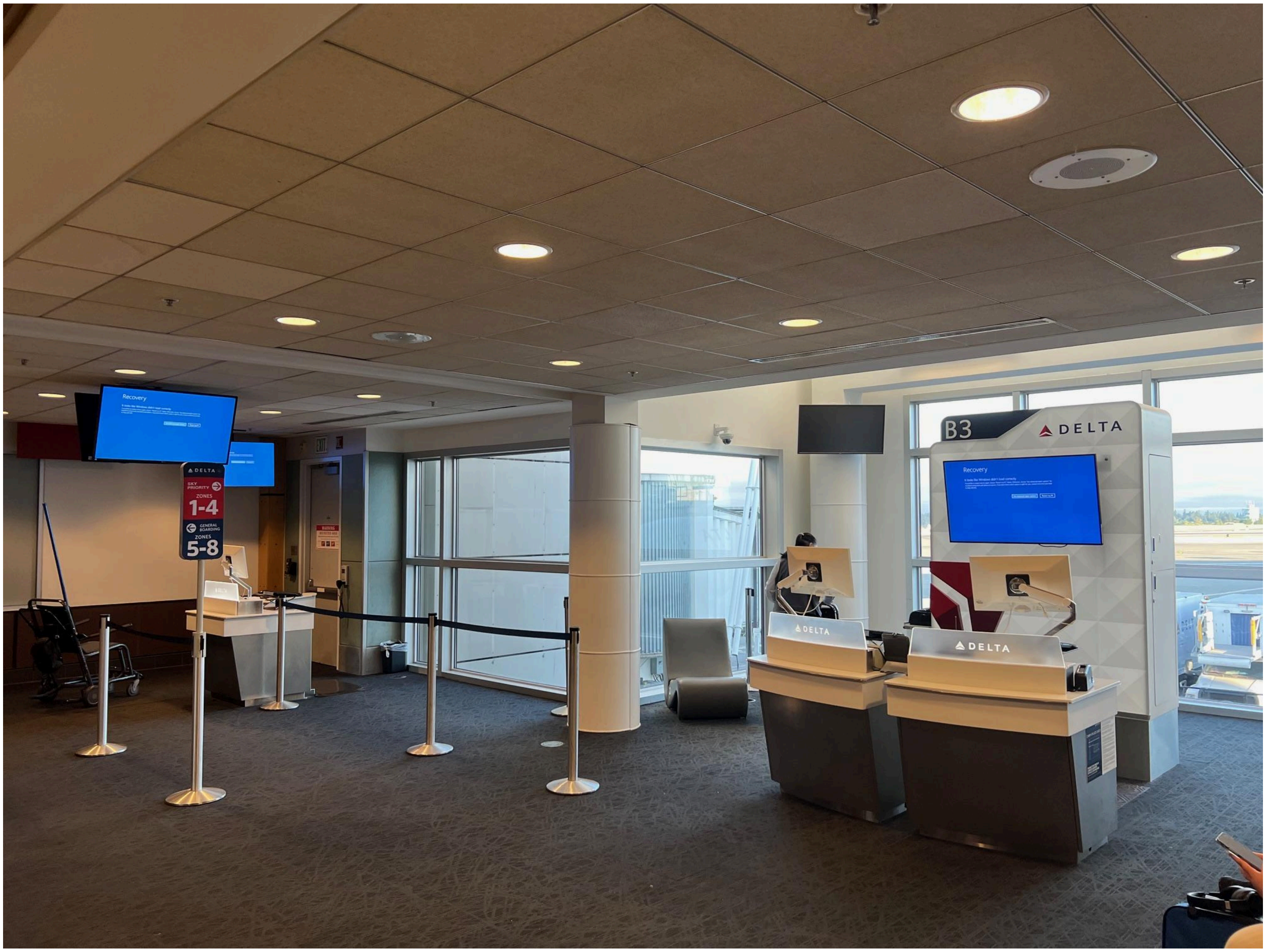
> *I am a week behind the podcast, finishing an audiobook. I am also late in sending this feedback so it may have already been sent to you by hundreds of listeners. :-)*
>
> *In episode 982, you talked about a Linux Daemon that would monitor logs and modify firewall rules to block IP addresses that made 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.*
>
> *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" the TCP connection handshake is not required. There can be a balance where the new firewall rules can have a time limit before they are removed, but if left to "block forever", a server can effective be disconnected from the Internet. — Kris*

Generically, Kris is absolutely correct that blocking incoming connections by IP opens up the possibility of creating deliberate denial of service attacks. But I’m unsure what he meant by writing “*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 3-way handshake has been acknowledged. It’s true that the client can provide data along with the final ACK in reply to the server’s SYN/ACK, but those round trips definitely do need to occur before the server’s TCP/IP stack decides that it has a valid connection. So while UDP services could definitely be spoofed to create such an attack, TCP-based services such as SSH, fortunately cannot be, so blocking based upon authentication failures would be spoof proof.

CrowdStruck

The Legend of Channel File 291

Blue screens of death at the deserted Delta terminal of the Seattle Tacoma airport, taken Friday morning by a listener since episode #1:



It's fortunate that GRC's incoming email system was in place and ready for this CrowdStrike event since it enabled a number of our listeners to immediately write last week to send some interesting and insightful feedback. These people do not hang out on Twitter.

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 can't access its data storage.*

Another listener wrote:

> *Good morning, I am new to this group ... only been listening for the last 8 years. I'm sure CrowdStrike will be part of next week's topics. 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.

Tom Jenkins (posted into GRC's newsgroup)

Crowd Strike 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.

This is 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. 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.

Seamus Marrinan (works for a major corporation which he asked me to keep anonymous)

For us the issue started at about 12:45 am Eastern time. We were responding to the issue by 12:55 and had confirmed by 1:05 am 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 am.

The order of recovery we followed were the servers, production systems, our virtual environment, and finally the individual PC's. In all there were about 500 individually affected systems across a 1500 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.

I'm quite impressed by the performance of Seamus' team. To be back up and fully running by 10am the day of, after 500 machines were taken down in the middle of the night is truly impressive. And I would imagine that whomever his team reports up 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. He took and sent photos of the sign on Dick's Sporting Goods the ***following*** day, Saturday, stating that they were closed due to a data outage.

A listener named **Mark Hull** shared this:

Steve, thanks for all you do for the security community, I am a proud spinrite owner, and have been an IT consultant since the days of DOS 3 and Netware 2.x. I do a lot of work in enterprise security, have managed Crowdstrike, write code and do lots of work with SCCM (MS endpoint management), as well as custom automation... so I feel I have a good viewpoint on the Crowdstrike disaster.

Crowdstrike is designed to prevent malware, and by doing so provide high availability to all our servers and endpoints. The fact that their very 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.

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. 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 this issue. Now it seems that corporations have eliminated this idea since this is not a revenue generating entity (they should research opportunity cost). With the onset of virtualization, I would argue the cost of this approach continues to decrease.

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 5, 10 and 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 5 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 for the first 1000 that report back would get the update and then the update would be paused before getting and evaluating this data.

It appears that either Crowdstrike doesn't have a design of this kind, or any internal testing, or that those were somehow bypassed in this case.

*I work with a customer that has Crowdstrike on every server, and all Windows desktops. The only servers that were **not** impacted were virtual machines on hosts that blue screened before it hit their VMs. In this case the carpet was yanked out from under the VM, and this can cause disk corruption. Of our 300 servers we did have to restore 3 from backups. Now desktop technicians will be visiting and manually restoring thousands of desktops. And the recovery task is more complex on machines where the drives are encrypted.*

In my opinion, the fact that this could happen must have been neglect on the part of crowdstrike. The stance of any good security company is to always think in terms of "What can go wrong", which they must not have done. What would happen if someone compromised their systems and used their global update process to push out something malicious. They have to balance getting new protection out to clients with ensuring that the “medicine is not worse than the disease” and in this case they failed.

I know that we need to protect software vendors from honest mistakes, but until they have some financial obligation for security breaches or this type of negligence, it is simply more financially beneficial to offer free credit monitoring or live with the consequences, than to pay to do the security right in the first place. I get this from some companies, but not from security companies, that are focused on endpoint high availability.

> *Finally, think of the hours and cost of those hours for each company involved. In our case technicians visited thousands of machines, not counting the number of companies whose business completely stopped during this tragedy. This along with the AWS outage of years ago may be the most costly IT issue of the decade. Thanks, Mark.*

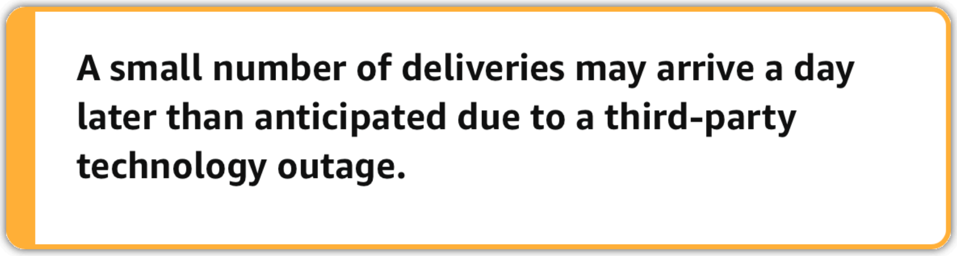
Samuel Gordon-Stewart in Canberra, Australia, wrote:

> *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 on 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 working register. Train services were halted as the radio systems are 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 update 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 had a chance to receive the update! Nothing will get fixed until the Head Office looks at it, but I think they'll be pleasantly surprised that only a couple devices need fixing and not dozens or more. Not my problem or role these days… although I did foolishly volunteer to help!*

And I saw this from the Amazon app on my iPhone:



A small number of deliveries may arrive a day later than anticipated due to a third-party technology outage.

Meanwhile, while in the U.S. almost all airlines were grounded with all of their flights cancelled, one was flying the friendly skies all by itself for the day. Digital Trends, reported under the headline *"A Windows version from 1992 is saving Southwest's butt right now"*, wrote:

> *Nearly every flight in the U.S. is grounded right now following a CrowdStrike system update error that’s affecting everything from travel to mobile ordering at Starbucks — but not Southwest Airlines flights. Southwest is still flying high, unaffected by the outage that’s plaguing the world today, and that’s apparently because it’s using Windows 3.1.*
>
> *Yes, Windows 3.1 — an operating system that is 32 years old. Southwest, along with UPS and FedEx, haven’t had any issues with the CrowdStrike outage. In responses to CNN, Delta, American, Spirit, Frontier, United, and Allegiant all said they were having issues, but Southwest told the outlet that its operations are going off without a hitch.*
>
> *Some are attributing that to Windows 3.1. Major portions of Southwest’s systems are reportedly built on Windows 95 and Windows 3.1, which is something the company has come under fire for in the past several years. It should go without saying that Southwest needs to update its system, but in this case, the ancient operating system seems to be doing the airline some favors to avoid a complete Y2K-level apocalypse.*
>
> *If you aren’t flying Southwest, you’re out of luck right now. Airports around the world had their scheduling systems crash in the wake of the CrowdStrike update, sending millions of travelers into a frenzy. The Federal Aviation Administration (FAA) said it’s working with several airlines on the outage. Thankfully, the FAA itself hasn’t been affected.*
>
> *Microsoft, who has been at the center of this fiasco with CrowdStrike, says that the root cause of the issue has been fixed. It could take days before everything is sorted out, though. Microsoft’s CEO Satya Nadella commented on the issue on X (formerly Twitter), saying, “We are aware of this issue and are working closely with CrowdStrike and across the industry to provide customers technical guidance and support to safely bring their systems back online.”*
>
> *That shows the scale of this problem. Microsoft has outages all the time, but none of them are worth commenting on from the CEO of the company. This is a different beast entirely, affecting millions of servers running on Windows. Southwest seems to have saved itself from any trouble by being woefully late to upgrade.*

I’ll just note that since this was not a Windows problem, but a 3rd-party problem, it was not companies running old versions of Windows but companies who were customers of CrowdStrike. That said, I’m certain that CrowdStrike cannot be used with Windows 3.1 or 95.

I was initially going to share a bunch of TechCrunch’s coverage of this. But then yesterday, Catalin Cimpanu (‘ka -ta-lin Sim-’pa-new), 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 into today’s podcast. I’ll get to that bit in a minute. But first, here’s Catalin’s summary:

> *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 (BSOD). 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 out.*
>
> *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 (eg. 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, even Russia had to deal with some outages.

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 horrible 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 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 are able to gain short 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. (extremely rough estimation, but probably in the correct range)

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. [eyeroll]

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 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.

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 (EDR) client with new techniques abused by threat actors.*
>
> *In this case, it was "Channel File 291" (C-00000291*.sys) that caused the crashes.*
>
> *CrowdStrike says this file was supposed to update the Falcon EDR to detect malware that abuses Windows named pipes to communicate with its command and control (C2) 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, 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 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. US 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, but I smell some politicians waiting to pounce on it with some "ideas." [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 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.*
>
> *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, 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. It's honestly one of the best and easiest phishing opportunities they've had in a while.*

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:

> *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. These businesses include large corporations, hospitals, transportation hubs and government departments. Most consumer devices do not run Falcon and are unaffected by this outage.*
>
> *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, 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.*
>
> *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.*

I haven't 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 out in the field. And we need that visibility.

Of course, 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 are 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 is 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 and via the CrowdStrike blog. Please continue to visit these sites for the latest updates.*
>
> *We have 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. Let's being with CrowdStrike's predictably (not very) technical update which they titled: *"Falcon update for Windows hosts technical details."* They open with the question:

> *What Happened?*
>
> *On July 19, 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) on impacted systems.*
>
> *The sensor configuration update that caused the system crash was remediated on Friday, July 19, 2024 05:27 UTC. 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, July 19, 2024 04:09 UTC and Friday, July 19, 2024 05:27 UTC, may be impacted.*
>
> *Systems running Falcon sensor for Windows version 7.11 and above that downloaded the updated configuration from 04:09 UTC to 05:27 UTC – were susceptible to a system crash.*

> *Configuration File Primer*
>
> *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 discovered by CrowdStrike. This is not a new process; the architecture has been in place since Falcon’s inception.*
>
> *Technical Details*
>
> *On Windows systems, Channel Files reside in the following directory:*
>
> *C:\Windows\System32\drivers\CrowdStrike\*
>
> *and have a file name that starts with “C-”. 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 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 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. 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.*
>
> *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.*
>
> *Root Cause Analysis*
>
> *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-**de**pendent global operational existence if around 1% – or about 8.5 million – extra-specially secured and running Windows OS-based machines 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 5-shots of espresso Starbucks. That’s 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 recovering 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 files – which Microsoft themselves examined, approved and digitally signed, was given the highest honor of being flagged as a "boot-start" device. Microsoft's 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 has the "boot-start" designation, 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 anti-malware defenses. When it is 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 Windows. 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 be to do its job.

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 inter-process communications. It's pretty much *the* way clients obtain services.

The on-the-fly code-signing service I created before the release of SpinRite v6.1 is started by Windows after it finishes booting. When that service starts up, 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 interconnection which allows the service and its client to 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 0x9C which is decimal 156. That value of 156 is loaded into the CPU’s r8 register where it’s 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 and declares an emergency.

What I’ve just described is a very typical series of events which precipitates what the entire world has come to know as the Blue Screen of Death. It’s something that should never happen, but hardware is not perfect and neither are people. So things that are not supposed to ever happen, happen anyway.

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.

We do not know exactly where that aberrant value of 156 came from. There’s been speculation about it looking like an offset into a structure, so that if the structure’s pointer were null, an offset like this might result and then be fed to a function that was expecting to be pointed to a member of a valid object. Some have observed that CrowdStrike’s channel file contains nulls.

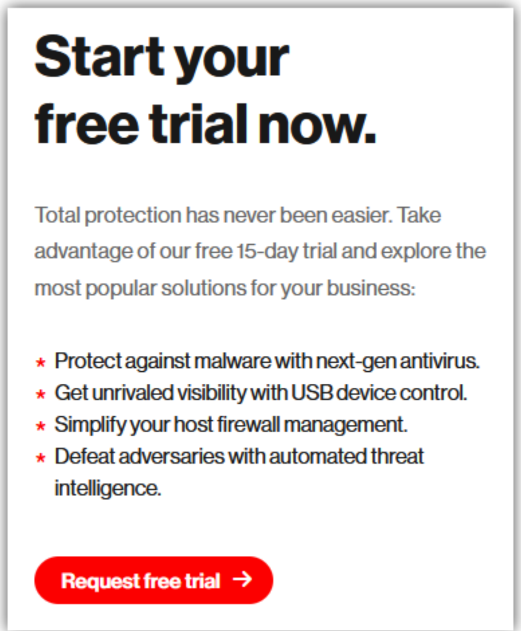
But the unsatisfying reality is that all we have today is conjecture. There is no question that CrowdStrike definitely and absolutely already knows exactly what happened, but they're not saying. I have absolutely zero doubt that their corporate attorneys clamped down the cone of silence over CrowdStrike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

There is very likely good news on that front, however. CrowdStrike's driver and its associated "channel file 291" that 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. It won't be forthcoming from CrowdStrike, but it won't need to.

So 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 File 291 to ever be deployed in the first place? 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 neck exposed.

I have a theory about this.



Start your free trial now.

Total protection has never been easier. Take advantage of our free 15-day trial and explore the most popular solutions for your business:

- Protect against malware with next-gen antivirus.
- Get unrivaled visibility with USB device control.
- Simplify your host firewall management.
- Defeat adversaries with automated threat intelligence.

Request free trial →

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 actually **not** 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.

I understand that to someone who does not code (like those in the US 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 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.

Note, also, that we don't know 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 from 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. So distribution failure should not explain this trouble, either.

I'll finish today with the acknowledgement that what has happened is so bad that I doubt we're going to soon learn the truth from CrowdStrike. They 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 George Kurtz will be delivering in front of Congress.