SECURITY NOW!

Transcript of Episode #880

RetBleed

Description: This week we start with a quick update on last week's Rolling Pwn problem. Then we look at the state of IPv4 space depletion and the rising price of an IPv4 address. We have an interesting report on the Internet's failed promise, Facebook's response to URL-tracker trimming, Apple's record-breaking Lockdown Mode bounty, Clearview AI's new headwinds, a new feature being offered by ransomware gangs, the return of Roskomnadzor, last Tuesday's patches, and some feedback from our listeners. Then we look at the details of the latest way of exfiltrating secrets from operating system kernels thanks to insecurities in Intel and AMD micro-architecture implementations. Yes, some additional bleeding.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-880.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-880-lq.mp3

SHOW TEASE: Coming up on Security Now!, it's me, Jason Howell, sitting in for Leo Laporte, who is on a cruise at the moment. So you won't see Leo this week. You will, though, see the man of the hour that is Steve Gibson, talking this week about Facebook encrypting its link URLs, incentives for cracking iOS Lockdown Mode, actually some pretty big incentives, so why don't you see what you can do. You make some money in the process. Clearview AI and how it's meeting total resistance around the world for the most part. And the bleeding continues as Steve dives deep into RetBleed. All that and more coming up next, Steve Gibson explaining it all on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 880, recorded Tuesday, July 19th, 2022: RetBleed.

It's time for Security Now! with Steve Gibson, the man of the hour. I'm Jason Howell filling in for Leo Laporte, who is on a cruise ship right now. Who knows what Leo's up to? But I'm here with Steve to hang out once again and talk security. How are you doing, Steve?

Steve Gibson: Let's hope he's not getting COVID. That would be a good thing.

JASON: Yeah. That's kind of top of the list of what we hope isn't happening. But let's hope he is having a good time.

Steve: Having a good time.

JASON: Meeting lots of awesome people. Eating good food.

Steve: But not too much.

JASON: We'll focus on the positive.

Steve: So we're at Episode 880 for here the middle of July. And Jason, I don't know what it is, but whenever you're on the podcast, we're talking about bleeding. And, you know, three weeks ago when Leo was off on the East Coast somewhere we had Hertzbleed. And now you're back and we've got RetBleed, which is...

JASON: Surprise.

Steve: ...subject and topic of the podcast. So another...

JASON: I bring the bleed. I don't know what to tell you.

Steve: Yeah, you bring the bleed. Another interesting side-channel attack on Intel and AMD processors. And this one has an interesting back story because Intel was sort of telling people that the way you're fixing this isn't really good enough. But then they decided, well, okay, but it does help performance, and we care about that. So we're just not going to say anything. Anyway, we're going to get to all that. But first we're going to briefly look back at last week's Rolling Pwn problem. Then we're going to look at the state of IPv4 IP space depletion and the rising price of an IPv4 address. We have an interesting report on the Internet's failed promise, which I thought was really sort of sad, but it's good some people are acknowledging that, well, it didn't really work out the way we hoped.

We also have Facebook's unsurprising response to URL tracker trimming, which was a subject last week. It didn't take this very long to drop. And they clearly had it in the works because it would have been complex for them to do, but they did. We've got Apple's record-breaking Lockdown Mode bounty. Clearview AI's new headwinds that they're facing. A new feature being offered by ransomware groups, three of them so far, but it's going to catch on. We've got the return of Roskomnadzor. Also last Tuesday's patches and some feedback from our listeners. Then, as I said, we're going to take a look at the details of the latest way of exfiltrating secrets from operating system kernels, thanks to what amount to insecurities in Intel and AMD's microarchitecture implementations. So yes, Jason, you're here, so we're going to talk about bleeding.

JASON: Another fine podcast with bleeding somehow making its way in. I don't know. I didn't do this on purpose, Steve. It's just the way the cookie crumbles sometimes. And we thank Steve in advance for the Picture of the Week, which is coming up right now. What you got?

Steve: So, okay, this is another one of those, it's sort of a variation on the theme of the path out in the middle of a large field that has a gate, like, across the path. And in the case of the path, because I love it, because there's also then, like, well-worn side paths, like just going around this gate. It's like, what? What? You know, what?

Anyway, so this one, though, is not that. This is somebody who locked up, and I use that term advisedly, yeah, they locked up their very expensive and nice-looking e-bike to a post. But, you know, it's a cylinder. And there's nothing on the top of the cylinder to prevent the person from just lifting the whole thing up in the air, off the cylinder. Like, I mean, so it's got the expensive bolt cutter-proof everything cord, cable cord thing, wrapped around the cylinder. Not tightly, either.

JASON: No, very loosely, actually, it looks like.

Steve: Yeah, they're very casual about this. So, okay, good luck. This has the advantage, unlike packets in the ether, that a thief would have to be very visible while they were lifting this bike up off of the cylinder that it's not really locked to. But anyway, the caption I put on this was "Hmm..." because, you know?

JASON: Yeah, this doesn't take too long to crack the code. John actually just whispered in my ear and points out that the weight of the battery, at least that's a little bit of possibly a preventative measure for lifting it up. You'd have to be a weightlifter to maybe lift that up compared to other people. Perhaps.

Steve: Yes, it did occur to me that were this like a motorcycle, which no human could lift - well, maybe Schwarzenegger. But, you know...

JASON: This would probably lift over pretty easily, though.

Steve: Yeah, yeah. And also for what it's worth the helmet is not secured. It's just hung with its strap on one of the handlebars. So, I mean, maybe the person - you could say maybe the person isn't far away. In which case, why lock it at all?

JASON: Yeah, yeah.

Steve: I don't understand the story here. Maybe they don't like their helmet?

JASON: Yeah, in general I think this person just has a cavalier attitude around security. It's kind of like, yeah, I mean, you know, it's almost like - it's like putting the sign in your front door that says we are protected by this security system when you actually aren't. It's like, you know, that's going to prevent some people from doing something. Like maybe some thief is going to look at this and be like, well, there is a lock there, and that's more trouble than another thing, so I'm just going to move on to another thing. But there's always someone out there that thinks it's worth the trouble. I looked at this, and I was like, you know, I've almost done this a few times, but I never actually followed through. It was like, well, I've got to put it to something. Should I? No, it really doesn't do anything.

Steve: You do wonder what this person's password is, though, don't you.

JASON: Yeah, probably pretty easy.

Steve: And they only have one, and they use it everywhere, and it's probably not that difficult to figure out.

JASON: It's "ebike." Ebike. They use ebike everywhere.

Steve: That's right. Or "ilostmybike" is their password. That's right.

JASON: Yeah, right, someonestolemybike, someonetookmybike.

Steve: Okay. So following up to last week's Honda-centric story, where the Honda engineers made the mistake of allowing their system, which resynchronizes their autos to their keyless remotes, by allowing them to move back to a previous state, which should never happen. Resyncing would have been fine if the resync was only allowed to move forward to a later state, which is actually all that should ever be necessary anyway. There's no safe way to allow an earlier state to be restored.

Anyway, last week when we covered this, the spokesperson for Honda told The Record, who was doing some follow-up reporting - and I love this, too. It hadn't really occurred to me. They said: "Hackers would need sophisticated tools and technical know-how to mimic remote keyless commands and gain access to certain vehicles of ours." Okay, well, it didn't hit me until just now as I was writing this that that statement makes no sense. If hackers did not have sophisticated tools and technical know-how to mimic remote keyless commands in the first place, then no rolling codes of any sort would need to be used at all. It's specifically because hackers do have sophisticated tools and technical know-how to mimic remote keyless commands that it's necessary in the first place to design a system with rolling codes, which Honda has failed to securely do, for the purpose of defeating hackers who had sophisticated tools and technical know-how to mimic remote keyless commands.

But in any event, that's not why we're back here this week. In addition, there was a dialogue which was spurred by last week's revelations. Honda said: "Honda regularly improves security features as new models are introduced that would thwart this and similar approaches." And then the spokesman added that all, and they said "completely redesigned," and I'm not sure what that means, completely redesigned 2022 and 2023 model year vehicles have an "improved system" that addresses the issue. Then they said: "Currently this includes 2022 Civic, 2022 MDX, and 2023 HR-V," saying, "Our newer system transmits codes that immediately expire, which would prevent this type of attack from being successful."

Okay, now, I think there seems to be some miscommunication somewhere because what's confusing is that the original hacking team used their system to crack 10 Hondas, with four of them being year model 2022, and one of those four being the Honda Civic, which this spokesperson claims has fixed the problem by using advanced technology. Like it's been fixed. But, you know, also note that all rolling codes immediately expire. That's the whole point of having them roll. They're inherently meant to be single-use codes that somebody can't capture and immediately repeat.

So the good news behind all of this is that hacking cars is fun, and doing so is an easy means to generate headlines, which really is the only payoff that most researchers seek or receive; right? I mean, they just - I often wonder, why did they spend so much energy doing this? But it's apparently for a little bit of your moment in the sun, and then onto the next hack. Since the hardware required to do this car hacking is now available inexpensively off the shelf, just put SDR or maybe "software defined radio" into Amazon's search and you'll get some.

So we can be pretty sure that automakers' past laziness with regard to their autos' true security will no longer go unnoticed and will be making future headlines whenever and wherever it is found to be lacking. And that as a consequence of seeing that happen a few more times maybe, they'll actually, like, figure out their communication, if nothing else, because Honda is still way messed up in that regard.

Okay. We've had a lot of fun through the years watching the saga of diminishing IPv4 address space. According to SIDN, which is the Netherlands' official domain registrar, IPv4 space address price, that is, the price of individual IPv4 addresses, has doubled in the past year. Back in 2015, so seven years ago, IPv4 space was selling around $5 per IP. This time last year that $5 had grown to between 25 and $35. That's last year. Today, a year later obviously, we're at $50 to $60 per IPv4 space.

Contrast this, of course, to IPv6 where there's essentially no practical limit, no practical limit to address availability. IPv6 addresses are not only free, but they are so freely available that ISPs hand out large chunks of IPv6 space to each of their residential subscribers. In fact, routing tables will not route individual IPv6 addresses. They won't. They can't. By definition there are too many of them. So they are only allocatable in big blocks. And so that's the way they're being allocated.

And I guess my point is the idea that here we have IPv6, and it is here. It's been here for quite a while. It's been defined for decades. Nobody wants it. They're willing to pay $60 for an IPv4 address rather than use free ones because, well, you know, those are weird. So it's probably difficult to find a better example of an entrenched unwillingness to change, to adapt, than for IPv4 space to be selling at this kind of premium when you can have all the 6's, the v6's that you want for free.

And yes, everyone, I know, ShieldsUP! doesn't do IPv6 yet. It's not that I'm unwilling to, it's I would love to. But I was distracted by SQRL for seven years, and now I'm back to SpinRite where I should be. Everyone agrees that's more important than having ShieldsUP! be IPv6 compatible. Once SpinRite has caught up to all the hardware platforms it needs to run on, that'll be, absolutely, that and the DNS Benchmark, everybody wants that to be IPv6, too. So yes, I've got to get my own house in order. I understand that. Everything's working except SpinRite. So that's the top priority.

Okay. I love this next piece. I titled it - or did they title it? Somebody titled it. Oh, yeah, they did: "Confronting Reality in Cyberspace: Foreign Policy for a Fragmented Internet." That's the official title of this huge 116-page report where apparently they were being paid by the page. A pull-quote from the article headlines the Executive Summary. It says: "The utopian vision of an open, reliable, and secure global network has not been achieved and is unlikely ever to be realized. Today the Internet is less free, more fragmented, and less secure."

Okay, now, I'm not going to drag us obviously through 116-page report. And although the report is obviously U.S.-centric, having been assembled by a U.S. think tank - who is it? Oh, yeah, the Council on Foreign Relations put this together. I think that everyone will find this interesting. I mean, if not a little sad and sobering. So here's just the Executive Summary from the report, which sums it up.

They wrote: "The global Internet - a vast matrix of telecommunications, fiber optics, and satellite networks - is in large part a creation of the United States. The technologies that underpin the Internet grew out of federal research projects, and U.S. companies innovated, commercialized, and globalized the technology. The Internet's basic structure - a reliance on the private sector and technical community, relatively light regulatory oversight, and the protection of speech and the promotion of free flow of information - reflected American values. Moreover, U.S. strategic, economic, political, and foreign policy interests were served by the global open Internet. Washington long believed that its vision of the Internet would ultimately prevail, and that other countries would be forced to adjust to or miss out on the benefits of a global and open Internet.

"The United States now confronts a starkly different reality. The utopian vision" - and here is the pull quote I said - "of an open, reliable, secure global network has not been achieved and is unlikely ever to be realized. Today, the Internet is less free, more fragmented, and less secure. Countries around the world now exert a greater degree of control over the Internet, localizing data, blocking and moderating content, and launching political influence campaigns. Nation-states conduct massive cyber campaigns, and the number of disruptive attacks is growing. Adversaries are making it more difficult for the United States to operate in cyberspace. Parts of the Internet are dark marketplaces for vandalism, crime, theft, and extortion.

"Malicious actors have exploited social media platforms, spread disinformation and misinformation, incited disparate forms of political participation that can sway elections, engendered fierce violence, and promoted toxic forms of civic division. At the same time, the modern Internet remains a backbone for civilian critical infrastructure around the world. It is the main artery of global digital trade. It has broken barriers for sharing information, supports grassroots organization and marginalized communities, and can still act as a means of dissent under repressive government regimes.

"As the Internet of Things (IoT) expands in coming years" - god help us - "the next iteration of the Internet will connect tens of billions of devices, digitally binding every aspect of day-to-day life, from heart monitors and refrigerators to traffic lights and agricultural methane emissions. The United States, however, cannot capture the gains of future innovation by continuing to pursue failed policies based on an unrealistic and dated vision of the Internet. The United States needs a new strategy that responds to what is now a fragmented and dangerous Internet. The task force believes it is time for a new foreign policy for cyberspace."

The major findings of the task force, which are then basically documented and substantiated by the remaining 115 pages, are - and we have a number of bullet points: "The era of the global Internet is over. U.S. policies promoting an open, global Internet have failed, and Washington will be unable to stop or reverse the trend toward fragmentation. Data is a source of geopolitical power and competition and is seen as central to economic and national security. The United States has taken itself out of the game on digital trade, and the continued failure to adopt comprehensive privacy and data protection rules at home undercuts Washington's ability to lead abroad. Increased digitization increases vulnerability, given that nearly every aspect of business and statecraft is exposed to disruption, theft, or manipulation.

"Most cyberattacks that violate sovereignty remain below the threshold for the use of force or armed attack. These breaches are generally used for espionage, political advantage, and international statecraft, with the most damaging attacks undermining trust and confidence in social, political, and economic institutions. Cybercrime is a national security risk; and ransomware attacks on hospitals, schools, businesses, and local governments should be seen as such.

"The United States can no longer treat cyber and information operations as two separate domains. Artificial intelligence and other new technologies will increase strategic instability. The United States has failed to impose sufficient costs on attackers. Norms are more useful in binding friends together than in constraining adversaries. And indictments and sanctions have been ineffective in stopping state-backed hackers."

So they conclude: "The task force proposes three pillars to a foreign policy that should guide Washington's adaptation to today's more complex, variegated, and dangerous cyber realm. First" - I'm going to turn down my email notifications. "First, Washington should confront reality and consolidate a coalition of allies and friends around a vision of the Internet that preserves to the greatest degree possible a trusted, protected international communication platform.

"Second, the United States should balance more targeted diplomatic and economic pressure on adversaries, as well as more disruptive cyber operations, with clear statements about self-imposed restraint on specific types of targets agreed to among U.S. allies. And third, the United States needs to put its own proverbial house in order. That requirement calls for Washington to link more cohesively its policy for digital competition with the broader enterprise of national security strategy."

So this is obviously what this podcast has been talking about for the last 17 years. We've been watching this happen. And I would argue that the Internet happened, and it wasn't very pervasive; right? I mean, it wasn't mission-critical. It was an interesting global communications platform. But it was as it inevitably became what it has become and that we've all watched over the last couple decades, its nature changed. It became important. It became something you couldn't do without. Communication, more and more communications moved to it. And it got to a point where control over it became something that everybody wanted.

It was supposed to be free and open and utopian, and everybody gets to talk to everybody, and countries that try to restrict it are going to crumble because you can't restrict it. Well, turns out you can pull the plug. And, whoops, no more Internet. We've had stories in the last few weeks we've been talking about where oppressive regimes are actually shutting the Internet down during national testing days because too many kids cheat and, like, use the Internet to do that. So, yeah, let's turn it off. So, okay.

The Executive Summary finished, listing 16 major recommendations. And just a couple of them stood out to me as being worthy of note. I've got five of them. They said for their recommendations: "Agree to and adopt a shared policy on digital privacy that is interoperable with Europe's General Data Protection Regulation," the infamous GDPR.

Now, that's interesting because we haven't done that in the states. And from our perspective, the GDPR, you know, it's kind of a mixed blessing; right? It's the reason we're having to say yes, dammit, I mean darn it, I accept these cookies, or do with cookies whatever you will. Or whatever. I mean, it's sort of created a mess.

Okay. Second major recommendation: "Declare norms against destructive attacks on election and financial systems." Okay, well, good luck with that. Third: "Negotiate with adversaries to establish limits on cyber operations directed at nuclear command-and-control communications systems." And obviously they would be bidirectional agreements, so we won't attack your nuclear reactors if you don't attack ours, and hold each other to that. Fourth, "Hold states accountable for malicious activity emanating from their territories."

And that's interesting because we've seen them say, well, okay, so the IPs were in our country, but we didn't do it. It must have been bad guys bouncing packets off of systems that they compromised. And so the point is, okay, you're still going to be responsible. If malicious traffic and activity comes from your country, then you need to be responsible for it. You're certainly responsible for restricting communications within your country, so you should be able to restrict malicious traffic coming from it equally.

And finally, clean up U.S. cyberspace by offering incentives for Internet service providers and cloud providers to reduce malicious activity within their infrastructure. And I thought that was interesting. We've talked about how DDOS attacks traditionally spoofed their IP addresses. That's happening less now that those sorts of attacks are less effective and are more easily blocked. But it always was the case that ISPs were allowing traffic to exit their control having IP addresses that did not exist within their borders. So it had to be spoofed. And it would have been trivial to have ISPs block that. But we're all one big happy Internet. So no such regulations were ever imposed.

So anyway, I just thought this was a really interesting report. I'm not going to go into it any further. But as I was scanning through it and reading some of the many other interesting details, I kept thinking that our listeners would really find some of the report's details interesting. So it is this week's Shortcut of the Week, so grc.sc/880, for anyone who's interested. It's a big PDF. But, boy, I think it's really interesting that this group has assembled a report that sort of formally states what the rest of us have all seen. And that is that, well, it was a nice idea, but didn't quite work out the way we hoped. So we need to, like, acknowledge that reality and figure out what we're going to do about it. Because if we keep doing nothing and just sitting around hoping, that's not going to turn out well, either. So grc.sc/880.

JASON: Didn't turn out quite the way we hoped. I feel like that could be on a T-shirt for the 2020s, you know, kind of the decade that we're in; right? It was a great idea, but didn't quite turn out the way we hoped.

Steve: Yes. While we're in a massive heat wave right now that is melting everything down. It's like, well, how did that 21st Century go?

JASON: Everything seemed like a really great idea at the time.

Steve: I hope the kids like the heat, yeah.

JASON: Yeah. All right, Steve. What's going on with Facebook? I feel like Facebook was like it. It was like everything Facebook. And then things got a little quiet and everything. Are they doing something right, or are they doing something wrong right now?

Steve: Well, so, just last week we talked about how Firefox v102 had added a feature to strip some of the tracking information from URLs that it was going to be querying before handing them over to a web server, the idea being that it would be enforcing the privacy of its users in that way. So this is something that users had to enable. But when it was, when it had been enabled, a small set of URLs, domains and then specific tags in the URL, which Firefox had been trained to recognize and felt comfortable with altering on the fly, would be altered. And we noted last week that Firefox was apparently being conservative about what they were stripping from the URLs since the Brave browser was reported to be significantly more aggressive.

Now, while discussing this last week I commented that, although I loved the idea of removing tracking identifiers from URLs, the whole thing felt flaky and uncertain to me since modifying a link's URL is inherently trouble prone, which is no doubt why Firefox was being apparently conservative in the URLs they were modifying compared to Brave; and because it would be so easy for Facebook, for example, to change the token name of the value in the URL link. Then all browsers would need to update their URL exception handlers, and we'd be back into a cat-and-mouse game.

Well, all of that handwringing, with regard to Facebook at least, has been rendered moot because Facebook's links have suddenly transformed into opaque blobs. And really this should not be a surprise to anyone. It should have been obvious that Facebook would not be happy having anyone mucking around with their URL links. The composition of any URL is by definition entirely up to the creator of the URL. Way back in 1994, RFC-1738, whose lead author was the famous Tim Berners-Lee at CERN, made clear that a URL is inherently an opaque token that only needs to have any meaning to the server that receives it. Once upon a time, URLs tended to directly reflect the hierarchy of the receiving server's file system, or at least some piece of that file hierarchy. And that file system was often organized by a human in some reasonable structure. So the whole thing meant something.

But as pages became more and more dynamic, being assembled on the fly by server-side PHP, ASP, or JSP scripting code after querying a big backend database, the primary reason URLs have remained at all understandable to humans is that they have been, and they'll continue to be for some time, but they've been a source of signals for Internet indexing search engines. We'd like Google to learn something about a page's link from its textual content, so that's often been preserved. But we've increasingly seen URLs being cluttered with things like GUIDs, those globally unique IDs, which only have any meaning whatsoever meaning to a server-side process.

Amazon's URLs, for example, have a short code near the front, which is surrounded by long hyphenated descriptive strings which describe the product. All of that superfluous text is only there for search engines to pick up on. Amazon has no need for it and completely ignores it. Since those massive multiline Amazon URLs are annoying to share, one of my favorite tricks is to strip everything out of an Amazon URL other than an anchoring, it starts with /dp/ followed by the 10-character product ID. That's all you need. And that results in a very short Amazon URL that always works.

In any event, all of Facebook's content is obviously all being assembled on the fly, driven by code and a massive backend database. So the construction of their URLs has always, or at least for a long time, been arbitrary and in no way reflects anything other than whatever their code wants it to reflect. So Facebook apparently decided for whatever reason, and it should come as no surprise to anyone, that it was probably tired of having third-party browsers and add-on extensions that are supposed to be enhancing privacy messing around with its links, specifically stripping out tracking information that they wanted to stay there. So now no one who doesn't know how to unscramble or decrypt a Facebook link can see anything about what's going on. They have truly become long opaque tokens.

Now, since older pre-encrypted links, that is, from more than, like, from before this weekend, which is when this suddenly began to happen, since those links are still going to be around, probably forever, I'm sure that all incoming links are now being checked to see whether they are old-style in-the-clear format or this new opaque-blob format. If they're old, they're accepted as is. If they're obfuscated by this new encryption, they'll first be decrypted, then handled.

So it's clear, as it always should have been, that any anti-tracking privacy enforcement we're going to obtain will need to be created by policy and, you know, laws and mandates and so forth, not by technology. Because ultimately this is something that Facebook has total control over, and they've just exercised another little bit of that control.

We discussed previously Apple's official launch, well, actually announcement because it's going to be happening in iOS 16, their announcement of this very interesting new "Lockdown Mode" feature. And that was announced during this year's World Wide Developer Conference. I think this idea makes so much sense because it's the, you know, our phones have this insane level of "it'll do anything you could ever want it to" breadth of features. And many are often unneeded, many are unwanted, and as a consequence, most of them go unused. I mean, there's stuff in my iPhone I'm embarrassed to say I have no idea what it does. There are things that annoy me, like the three dots that are now at the top. I keep trying to pull down from the top, and now I'm getting the multitasking stuff, where I didn't before. That's annoying, and you used to be able to turn it off. Now you can't.

Anyway, these things are like crammed with features that most people don't need, don't want, and don't use, yet being there hugely increases any device's attack surface. There are just more things that could have parsing errors in them that could go wrong. So simply turning off all of that unwanted and unneeded excess for individuals, especially for individuals for whom security trumps the ability to receive cat videos from strangers, seems like an obvious win. And when Leo and I were talking about this last week, it seemed like something we would both be inclined to turn off. At least, you know, take it out for a spin and see how it affects our lives. It doesn't seem like it's going to be that restrictive, even though all the press talking about it is saying, "Oh, my god, it's super lockdown restrictive. You can't do anything anymore." I don't do anything anyway. So it's not going to affect me very much.

So anyway, my feeling is it will probably go a long way toward limiting the victimization by commercial malware such as Pegasus, which is explicitly Apple's target here. And because Apple thinks so, too, they've decided to, as they say, put their money where their mouth is by offering the industry's largest bounty ever $2 million to anyone able to reproducibly crack into an iOS 16 device when it's in Lockdown Mode. I say bravo. I think that's cool. And I'll bet that a bounty of that size will likely give those who used to just enjoyed jailbreaks for the fun of it some new incentive because $2 million. Wow.

We've talked a lot in the past through the years about Clearview AI. They're the company, just to remind anybody maybe who hasn't been listening for years, that decided what they would do is send bots out onto the Internet, much as Google sends spiders. They would send their own bots out to collect images from publically available social media, you know, crawling Facebook. Crawling Twitter. Crawling everything. And building a huge database of people's faces, which they would then, using other means, tie back to their location and their availability, and build a big database. That's Clearview AI. And it's been a big hit with law enforcement and governments and any entities that have some need to identify people from photos.

Okay. So Clearview AI has been in the news just recently. Essentially they've been fined by Greece's privacy authority, the Hellenic Data Protection Authority (HDPA), for violating parts of Europe's infamous GDPR. The fine which has been levied against Clearview AI by the HDPA is a hefty 20 million euros. And what's a little bit galling, even to me, is that it's not due to any use or abuse of Clearview AI's admittedly controversial facial recognition database technology. It's just because Clearview AI exists, and Greece doesn't like the idea. And the GDPR gives them the right to fine Clearview AI over their conduct, even though there's no implication of its use.

A 22-page decision demands that Clearview AI stop processing biometric data on individuals in Greece, and said the company must delete all the data, that is, all the pictures of Grecians, it has already collected. The decision stems from a complaint filed by a number of privacy organizations which questions Clearview AI's practice of scraping selfies and photos from public social media accounts as a means of assembling its facial recognition database, which is rapidly growing toward, well, actually I think it's at 10 billion, and they're trying to go to 100 billion.

Okay, now, as we know, since we've been tracking this interesting edge case since they emerged a number of years ago, Clearview AI sells - it's in the business of selling its facial recognition tools to law enforcement agencies around the world and has said they want to get to 100 billion images. It's also the case that Clearview has been at work in Ukraine, helping to identify both deceased Ukrainian citizens for the government, and Russian soldiers so that families can be notified back in Russia in case they want to come and pick up their dead Russian.

The problem that Clearview AI faces surrounds consent. More and more privacy regulations are requiring consent, but Clearview's autonomous image-scraping technology is inherently consent-free. What I thought was interesting is that while Greece's Hellenic Data Protection Authority has levied this hefty fine, Clearview AI has never, never had any contact with either Greece's citizens or its law enforcement agencies. They simply share the same planet.

Clearview AI said it does not have a place of business in Greece or the EU, and it does not have any customers in Greece or the EU. The company also claimed its product has never been used in Greece and "does not undertake any activities that would otherwise mean it is subject to the GDPR." One of the several privacy groups which filed the initial complaint explained that the fine and the ruling made clear that the GDPR is applicable because Clearview AI uses its software to monitor the behavior of people in Greece, even though the company is based in the U.S. and does not offer its services in Greece or in the EU. The privacy organization said: "Collecting images for a biometric search engine is illegal, period."

So one thing that made me just shake my head is that Clearview has made it clear that they're happy to steer clear of regions that don't want their services. Yet the Greek authority also ordered Clearview to appoint a representative in the EU, even though they don't want to do business in the EU and haven't and aren't, to enable EU citizens to exercise their rights more easily, meaning I guess they would like someone local to sue, and so regulators have a contact person in the EU. Yeah, I don't blame Clearview for not doing that.

So I don't mean to sound overly sympathetic toward Clearview AI. But this does sort of seem to be, I don't know, overreaching. All of the images it's collecting are public.

Anyone can view them. Just like the web pages that Google crawls across and indexes which allows us to later locate the information we seek. So it's clear that the difference is that pictures of people's faces are considered to be biometric data, even though faces are kind of public. It's considered biometric data by these regulators and regulations, and are not regarded any differently than fingerprints or DNA. If someone followed us around, dusting everything we touched to lift our fingerprints, that would likely annoy us. The fact that Clearview AI's image collection is unseen doesn't render it any less noxious in the eyes of privacy regulators.

One country after another is lowering the boom on Clearview AI. We previously talked about the U.K.'s 7.5 million euro fine last May, similar rulings have recently been made by France and Italy, and Austria is said to be preparing a similar ruling. So it's looking like maybe this U.S.-based company will actually only be able to operate in a country where its privacy laws do not exclude it from doing so because we know that Illinois and their PIPA, the state of Illinois, that's a problem because of PIPA, which is where some of the earlier suits have been filed. Now we've got lots of EU countries doing so under the GDPR. So it's looking like the territory that Clearview AI's going to be able to cover is shrinking, and it's actually looking like this is a fight it's going to lose.

And speaking of searchable databases, several ransomware and extortion groups have been creating searchable databases of information they have stolen during their attacks. As we know, it's not news that ransomware groups have been extorting organizations with the threat of leaking the data that they have stolen. They steal it, they exfiltrate it, then they encrypt it so the company that owns it can't have it. And also, adding insult to injury, they've got a copy of it, and they're threatening to release it publicly unless the ransom is paid. Well, now they've gone one step further and created, started indexing the data and making it searchable. Over the last month or so two ransomware groups, actually three - AlphV, Karakurt, and LockBit - have all debuted features on their leak sites which allow visitors to search through the troves of data by company name and/or other signifiers.

A senior staff researcher at Tenable has confirmed that all three groups have incorporated some kind of searchable database functionality into their leak sites. And if we've seen anything, it's that an idea that's useful will be quickly picked up and mimicked by other ransomware groups. So we can soon expect this to be a new feature of all the dark web exfiltrated extorted data leak sites.

Emsisoft's threat analyst Brett Callow said that the tactic was designed to further increase the pressure on organizations by weaponizing their customers and business partners. Callow said: "The gangs likely believe that making the data available in this way will result in more companies paying due to a perceived increase in the potential for reputational harm. And they may be right." He added that in the past, companies have been able to dodge accountability for the leaks by claiming that there is "no evidence user data has been misused," which is a line seen in hundreds of breach notification letters over the past few years. Callow notes that such "soothing statements like that aren't really possible when people know their personal information was exfiltrated, compiled into an individual downloadable pack, and made available online." Who knows, maybe Google will start indexing it, too.

Moscow has imposed a $358 million fine, $358 million, more than chump change, on Google over Google's continued failure - which I guess at this point you'd have to consider, you'd have to call it a refusal since here we are in July, and the attack on Ukraine was in February, as I recall - Google's failure to filter out information from its search results that Russia's Internet watchdog "Roskomnadzor" has demanded be removed. I should note that the amount of the fine is much more fun when expressed in Russia's much less valuable rubles. That would be a total of 21 billion very small rubles.

Anyway, Roskomnadzor announced that Google and its subsidiary YouTube have failed to remove the following materials after multiple requests: information about the course of the "special military operation" in Ukraine, which discredits the Armed Forces of the Russian Federation; content promoting extremism and terrorism; content promoting harmful acts for the life and health of minors; and information that promotes participation in unauthorized mass actions.

So as we see, a free and open Internet isn't always the best thing for everyone. I suppose this is what the Council on Foreign Relations meant when they said that the dream had not come true and the sooner we in the West, and the U.S. specifically, wake up and smell the packets, the better.

I guess Roskomnadzor realizes that Google is too useful to block outright, or they would have. They've tried over and over to enforce sanctions based on various parts of Russia's Code of Administrative Offenses. Last month, Roskomnadzor fined Google $1.2 million. That's a measly 68 million rubles. But as the fines remain unpaid, the multiple violations qualify it to be based upon a different practice, which is a piece of the action. In this case, up to 10% of Google's annual Russian revenue. Russian users of Google Search and YouTube will now also encounter a warning about Google's violation of the law, and the will not be allowed to place advertisements or use them as information sources.

So Russia is attempting to squeeze Google by the wallet. And, for what it's worth, it's working. Google's paid services are disappearing and being withdrawn. After Russia's invasion of Ukraine and the so-called "anti-fake news laws" which were enacted in Russia, which amounted to don't say anything we don't like, Russia's Google subsidiary, Google LLC, filed for bankruptcy, claiming it had no ability to continue business after a series of massive fines and, ultimately, asset confiscation. So loyal Russians will presumably think, well, that's just those corrupt Westerners getting what they deserve. On the other hand, they will no longer have access to Google services. And I suppose that was inevitable. I guess I would give Google a tip of the hat for not bowing to Russian pressure and doing their part to keep the Internet open.

And speaking of getting what we deserve, last Tuesday Windows users received patches to hopefully fix a total of 84 individual flaws across Microsoft's sprawling software base. One of those was a true zero-day privilege elevation bug which was being actively exploited in the wild. The demographics of the patches break down: Of those 84, 52 were elevation of privilege vulnerabilities; 12 allowed remote code execution; 11 supported information disclosure; five were denial of service vulnerabilities, meaning that something crashed; and four were generic security feature bypass, whatever that means. There were no reports of any big meltdowns following last Tuesday's updates, so nothing big and obvious was messed up this month. A handful of bugs are no more. Well, except for any new ones that may have been introduced. Maybe we'll get to those eventually, as well as all the others that still remain in Windows and Microsoft's other products.

Okay. So I got a DM'd tweet yesterday which sort of surprised me. It was a fun SpinRite testimonial, and of the sort we haven't heard for a long time. It came from a guy named Paul Jolley. He said: "Last week one of our power stations reported they needed to restore a GEM80 PLC." PLC is a programmable logic controller; right? He said: "They had two separate backups on 3.5" floppy disks, but neither would read. Configuration control," he wrote, "knowing what code is running on programmable devices performing process control in an OT environment is very important in our industry, so they were in a pickle. They tried a number of ways to read the floppies using various freeware and were unsuccessful. So I offered to try SpinRite as a last resort."

He says: "I took delivery of the floppies this morning and set version v5.0 to work on Level 2. It managed to recover about 90% of the file required from the first floppy. Then from the second floppy, which had a totally corrupt file system," he said, "I was able to 'cat' the entire device to Linux to a file and subsequently extract the same file contents. Combining the recovered data from both floppies provided full coverage, and thanks to you I was a hero."

So I think what he meant was that he recovered everything but 90% of one file, but that file he was able to recover from the other floppy. So given the two and SpinRite, he was able to succeed. What I found interesting was Paul's reference to SpinRite 5. As I've previously mentioned, for some confounding, apparently mystical reason, v5 is superior to v6 for the recovery of diskettes. I've stared at SpinRite 6's code for probably a total of days at this point, trying to explain the difference. There is no difference. So I have no idea why. But in testing both, v5 has consistently produced superior results.

The other thing that was interesting was that, by being a DM, I had our previous DM thread. It turns out back on February 13th of 2021, so about a year and a half ago, Paul had DM'd to ask. He said: "Happy to contact GRC support, but thought I could quickly ask you first. I listened to a recent podcast where you said SpinRite 6 owners could download SpinRite 5 if they want by simply changing the download URL. I was interested because at our site we still use floppy disks." And of course he's talking about for these PLCs controllers.

He said: "So I looked up my purchase email and followed the link to the download page where it asks for my transaction code, then generates links that don't have a version in the URL. I must be missing something or didn't understand what you meant on the podcast." Anyway, he later tweeted: "Was just about to contact Greg this morning when I found the answer was on the FAQ page at the bottom." And sure enough, down at the bottom we explain how to change the download URL to allow yourself to get a copy of SpinRite 5 because for reasons that will never be known at this point, I think it's safe to say, 5 is better than 6 at recovering data from floppies.

Okay. A couple closing-the-loop bits. Michael Swanson, he said: "Hi, Steve. I just listened to SN-879," so that was last week's podcast. He says: "And regarding the use of a VPN when traveling, or even at a coffee shop," he says, "I prefer to use a travel router like the TP-Link N300. I connect the travel router to whatever Internet service is available. And whatever devices I bring with me - laptop, tablet, phone, Roku, et cetera - connect to the WiFi network of the travel router. All my devices are then behind a full NAT firewall. Added security," he said, "the travel router is also using Google DNS to prevent DNS hijack, and it is also possible to set the router to be a VPN client to many VPN services, and thus tunnel through to any VPN exit point including my home network, if desired. And the WiFi network on my travel router has the same SSID as my home network so all my devices connect automatically thinking they're at home."

So Michael, thanks for sharing that. I thought that was very clever. I liked the idea of using the same SSID and obviously the same password so that when you're on the road your devices don't know that you're not home, and they connect easily. And obviously he also understands that to get the same security of a VPN, you still would need to use a VPN tunnel, although it definitely is nice to be behind a NAT firewall. If something in the hotel was trying to get into your devices by port scanning them, having a NAT-based firewall would also solve the problem.

However, there was an even cooler idea, I think. IcyvRan is his handle. He said: "Hi, Steve. One solution, if one does not trust a WiFi hotspot, is setting up a Raspberry Pi at home" - or whatever you want to actually - "with Tailscale, and configuring it as an exit node." And he provided a link, which I have in the show notes. The link is to an FAQ, an explainer page about Tailscale's exit nodes. First of all, remember that Tailscale, we've talked about before, is a so-called overlay network, very much like Hamachi was back in the days when 5-dot was unallocated Internet IPv4 space. That didn't last forever.

Anyway, about exit nodes, Tailscale says: "Exit nodes capture all your network traffic, which is typically not what you want. The exit node feature lets you reroute all your non-Tailscale Internet traffic through a specific device on your network. The device routing your traffic is called an 'exit node.' By default, Tailscale acts as an overlay network. It only routes traffic between devices running Tailscale, but does not touch your public Internet traffic, such as when you visit Google or Twitter. This is ideal, they wrote, for most people, who need secure communication between sensitive devices like company servers, home computers and so forth, but don't need extra layers of encryption or latency for their public Internet connection.

"However, there may be times when you do want Tailscale to route all your public Internet traffic - in a cafe with an untrusted WiFi, or when traveling overseas and needing access to an online service, such as banking, only available in your home country. By setting a device on your network as an exit node, you can use it to route all your public Internet traffic as needed, like with a consumer VPN." So I thought that was a very cool tip. So thank you, IcyvRan, for that. And just a heads-up for all of our listeners. Tailscale can do that.

Taskel tweeted: "FYI on the Quantum-Resistant algorithms CRYSTALS-Kyber and CRYSTALS-Dilithium." Remember that we talked about that recently, that the NIST had chosen four of the eight next-generation cryptographic algorithms that would be used to provide quantum resistant crypto. I loved Dilithium crystals because of course they power the warp drive on Star Trek. I didn't know what Kyber crystals were. Well, he writes: "Kyber crystals are what is used in lightsabers in Star Wars." So there was something for Star Trek and for Star Wars fans there. Thank you, Taskel. Didn't know.

And someone tweeting as Lethal Dosage tweeted: "I logged into Twitter for the first time in four years to poke fun at you. You are losing geek points. The first Star Wars movie was not Episode IV - A New Hope, it was just 'Star Wars.'" He says: "Episode IV - A New Hope was added later." And then he said: "Watch the original intro, only two minutes long." And he provided a clip, a YouTube link, to, sure enough, a two-minute capture of "Star Wars 1977 original opening crawl" is the title of it. It's had 1.4 million views, and it's there.

And I have to say it absolutely looked authentic. But it's old and grainy, and in this day and age it could easily have been edited. So I did a bit of digging around the Internet, and I got the whole story. In the beginning, there was just "Star Wars." But then fans of what turned out to be the most popular science fiction movie of all time were thrown a hyperspace curveball. The film known as just "Star Wars" turns out wasn't the beginning of the story. It was the middle. Four years after the original film hit theaters, it was re-released, this time being called "Star Wars: Episode IV - A New Hope."

So here's what happened. In March of '78, right, Star Wars the original movie was released in '77. Next year, March of '78, the science fiction author Leigh Brackett died, and George Lucas took over writing movie number two, which was titled "The Empire Strikes Back," a task which he shared with Lawrence Kasdan.

Next, Lucas decided that there's a bigger back story to all of Star Wars, which means that the Empire, you know, "Empire Strikes Back," is not Part II, but instead Part V. So in 1980 "The Empire Strikes Back" identified itself as "The Empire Strikes Back: Episode V," which totally blew everyone's mind at the time, resulting in no end of confusion. Then the next year, in '81, the original Star Wars movie was re-released as Episode IV to make everything line up properly.

And what's confusing about all this is that I definitely saw the original Star Wars in 1977. I mean, I was alive, so of course I saw it. I was 22 years old. So I recall still that afternoon, 45 years ago, sitting in the theater and seeing this movie with some friends that I worked with. And I distinctly also recall anxiety and consternation being created by Star Wars episode numbering. But I guess the anxiety must have been created when "The Empire Strikes Back" identified itself for the first time as Episode V, and it was like, what? Rather than when the original Star Wars identified itself as Episode IV, which it didn't in the beginning. So now we all understand that. And Jason...

JASON: That's interesting to hear that, yeah.

Steve: You obviously are not 45 years old.

JASON: Close.

Steve: You have always had Star Wars in your life.

JASON: I have always had Star Wars in my life. And I don't think that I ever really - you know, okay. So, well, I'm almost 47.

Steve: You are?

JASON: So I'm close to the age that you were then now.

Steve: This is the best moustache you can muster.

JASON: Yeah, this is all I can do, yeah. Believe me, nothing even grows right here. Anyway, that's beside the point.

Steve: Hey, I wish that were the case. That would be nice.

JASON: It kind of shaves itself, to be honest. But yeah, I've always had Star Wars in my life. I never really felt very weird about the numbering because Star Wars came out, yes, when I was too little to watch it. I think I was two at the time, when it was actually in the theater.

Steve: Two, yes.

JASON: And so the whole numbering thing was already in place by the time I was ever at all aware. But I do remember "Empire Strikes Back" and loving it. Back then that was my favorite of those three, anyways. But what this makes me think is like, I've only ever watched "Phantom Menace," which is the first one, right, one time, and I thought it was awful. This was right around the time it was in the theaters. I thought it was horrible. And so as a result I never gave Part II and III any chance. I never actually watched them. I've still to this day not watched them, and I don't know why. Maybe I need to do that. But I can't help but, like, hear what you just said, [crosstalk] watch them.

Steve: I do know why - [crosstalk] watch them.

JASON: Okay. See, there you go. So I hear what you just read on the podcast, and I'm like, all right, it all makes sense why those earlier ones were not very good because it was kind of an after-the-fact thing, like oh, wait a minute, there's more we can do here, but I'm not sure what that more is yet.

Steve: Yeah. And more wasn't necessarily better.

JASON: Yeah, it wasn't necessary.

Steve: We immediately descended into little teddy bears running around. And, you know, it's just like, what has happened?

JASON: Yeah, little strange, the arc.

Steve: So I have one last thing to share. Dave Pope, he said: "FYI, my 2013" - okay, 2013 - "Ford key fob has bidirectional comms. It has a light that shows me red or green if the remote start was successful or not." He says: "No idea if it does the handshake you mention in the episode, though." Because I talked last week about the only way to really secure remote keyless entry. Tesla does a better job than Honda because it'd be hard to do a worse job than Honda. They do a better job by only ever moving the synchronizing counter forward so that codes actually do expire the first time that they're used, and there's no way to trick the system into using the same code again. Although, as we saw, an active attack can use jamming in order to get the key fob to emit additional codes that aren't seen by the car, which an attacker can grab to use in the future. But the way to absolutely solve the problem is bidirectional handshake, which is what we have on the Internet for all of our secure comms, and that's robustly secure.

Anyway, I just thought it was very cool that, what, nine years ago Ford has a key fob that lights to let you know whether the car has affirmatively confirmed that the car - that the engine started or not. That's very cool. So anyway, David, thanks for sharing that. And Jason, let's tell us about our last sponsor, and then we're going to go to back into bleeding because you're here on the podcast.

JASON: Oh, boy, yeah. I hope next time I join it has nothing to do with bleeding. Let's just put that on the table right there. Let's talk a little bit about RetBleed. What exactly is it, and why are these things appearing every time I host the show?

Steve: Yes. So, okay. Ret is the universal name, I think it's like universal across all processors. I don't know if I've ever - I've programmed many chips in assembly language, and Ret has always been the name of the CPU instruction for causing a subroutine to return. So Ret is short for Return. It's placed at the end of a subroutine to cause the subroutine to stop at that point. So it's a little bit like - it's like a special jump instruction. It stops its execution, and it returns, it causes the processor to return to the instruction following the one that invoked the subroutine. So in essence the instruction tells the CPU to return to the point where the subroutine was called, like just after the point it was called. So execution resumes in a linear stream from that point.

In stack-based processors subroutines are often provided with some parameters which they will use for whatever work they need to do. So the caller puts these parameters onto the stack and then the subroutine looks on the stack in order to access them. They can be values or pointers or whatever. And subroutines may place some of their own local temporary data onto the stack, as well. And how many times on this podcast have we used the term "stack buffer overflow"? Uh-huh. Meaning that there was a buffer that some code had put on the stack, and it overflowed the stack. That's always been a big problem. And when the processor's return instruction is executed, all of this stack-based data is discarded. Nobody bothers to, like, flush it to zeroes or overwrite it because that takes time. Instead, the stack pointers just moved back above it as if it never existed, and we go on from there. So it's a very elegant means for managing various sorts of temporary data.

RetBleed is the brainchild of two researchers from ETH Zurich, who have been behind a number of previous very clever attacks. Their paper on RetBleed, which is what they named this, will be delivered in a few weeks from now, I think it's August. I don't remember now. I had the date in my head. It's gone. Anyway, a couple weeks from now during a Technical Session of the USENIX Security '22 conference. They, being good guys, they responsibly disclosed their discovery to Intel and AMD back in February of this year, presumably with a six-month non-disclosure period. They agreed to be silent. That embargo was lifted last Tuesday, the 12th of July, which also happens to be Patch Tuesday, when it turns out some fixes for RetBleed were pushed out to the world.

Okay. So I'm going to start by just reading their paper's abstract. I'm not going to get into the weeds because the weeds are very deep and thick here. But the abstract gives us an overall feel for what this is. And then I will break it down some.

So they wrote: "Modern operating systems rely on software defenses against hardware attacks. These defenses are, however, as good as the assumptions they make on the underlying hardware. In this paper, we invalidate some of the key assumptions behind retpoline" - I'll explain that in a minute - "a widely deployed mitigation against Spectre Branch Target Injection (BTI) that converts vulnerable indirect branches to protected returns. We present RetBleed, a new Spectre-BTI" - again, Branch Target Injection - "attack that leaks arbitrary kernel memory on fully patched Intel and AMD systems.

"Two insights make RetBleed possible. First, we show that return instructions behave like indirect branches under certain microarchitecture-dependent conditions, which we reverse engineer. Our dynamic analysis framework discovers many exploitable return instructions inside the Linux kernel, reachable through unprivileged system calls. Second, we show how an unprivileged attacker can arbitrarily control the predicted target of such return instructions by branching into kernel memory. RetBleed leaks privileged memory at the rate of 219 bytes per second with 98% accuracy on Intel Coffee Lake, and 3900 bytes per second with greater than 99% accuracy on AMD Zen 2 chips."

So, okay. There are a few things to observe here. One is that this is another instance of the lesson that attacks never get worse, they only ever get better. When we started off with the Spectre and Meltdown speculative execution attacks, they were purely theoretical. This was at the end of 2017, early 2018. It's all we were talking about. Purely theoretical. But they didn't remain that way for long. Before long researchers were discovering how to use these once-theoretical attacks to probe the contents of memory that they had absolutely no valid access to. That access limitation was enforced by hardware, and it didn't matter.

Essentially, they deliberately created a road that would not be taken by the CPU, but which the CPU would speculatively prepare to take anyway. And in doing so, it would preload the contents of some memory down that road into its cache. Then they would probe the cache to see what the CPU had cached in preparation for that never taken road. In this manner, they would get the CPU to access memory for them which they could not legally access themselves. Access violations were never triggered because speculation never triggers access violations. This all amounted to some extremely clever manipulations of the insanely complex microarchitectures that have been incrementally added to, generation after generation, to modern processors, all in the name of squeezing out every last cycle of performance.

What's annoying to researchers, who are just wanting to, like, make the world more secure, is that the microarchitecture is undocumented. It is never documented. Intel just says, oh, don't worry about it, it's perfect. Except it's not. And so the first thing these guys all have to do is painstakingly reverse engineer the underlying architecture in order to figure out how it works. That they can only do by observing performance in all kinds of crazy tests. They reverse engineer how this all works underneath the chip's instructions. Then they go about bypassing the protections in some instances that this system provides. An amazing amount of work. And, you know, what, they get a paper out of it. They ought to be rich.

Anyway, the problem that's the subject of this paper, and of much sudden scurrying around - for example, as I'll explain in a minute, Linus just delayed the next Linux kernel release by one week as a result of this - this has the name "branch target injection." It's also known as Spectre Variant 2. There are essentially two available mitigations for this sort of speculation side-channel leakage. There is retpoline, which is a contraction of return trampoline, thus retpoline; and IBRS, which stands for Indirect Branch Restricted Speculation.

Just over three years ago - oh, I should mention, IBRS, Indirect Branch Restricted Speculation, is Intel's official solution and has always been. Retpoline is what Google cleverly came up with quickly as a fix for Chrome because Chrome would have been a big target for this. It turns out that you could actually do this in a browser. And so Google had to fix the Chromium engine, hardening it against this Spectre Variant 2. Thus they invented retpoline, which people liked a lot better than Intel's IBRS solution.

Just over three years ago, the SUSE Linux blog posted an article titled "Removal of IBRS Mitigation for Spectre Variant 2." And what was written is interesting in light of today's events. SUSE wrote: "As the Meltdown and Spectre attacks were published at the beginning of January 2018, several mitigations were planned and implemented for Spectre Variant 2. Spectre Variant 2 describes an issue where the CPU's branch prediction can be poisoned, so the CPU speculatively executes code it usually would never try to. For instance, user space attacker-controlled code could make the kernel code speculatively execute Spectre code gadgets that disclose secret kernel information, via flush-and-reload" - those are cache timing - "disclosure methods."

They said: "Two major mitigations were proposed." That is, for Spectre Variant 2. "A CPU feature called Indirect Branch Restricted Speculation that would not use branch predictions from lower privilege levels to higher ones." Meaning when jumping into the OS from userland. They said: "Or software workarounds called 'retpolines' and 'RSB stuffing.'" They said: "These can fully replace the IBRS mitigation." Except not now. But that's what they said at the time. They said: "On Intel Skylake there is the theoretical possibility that these software mitigations are not sufficient, but so far research has not shown any holes." Well, of course that was true three years ago. But as we know now, it is no longer true.

They said: "SUSE backported the IBRS patches to our kernels" - meaning backported I'm sure from Linux - "to our kernels for the initial release of mitigations and enabled them, as the retpoline mitigations were not yet ready. SUSE pushed the retpoline mitigation some months later after support in the compiler and kernel became available, but left in the IBRS mitigation." Which they now wish they had left in. "As of today" - again, this was three years ago - "the retpoline and RSB stuffing software workarounds provide the same level of mitigations that IBRS provides. While IBRS support continued in the SUSE kernel, it was not accepted by the Linux upstream kernel community, and it was also shown to cause performance degradation." And how.

Finally, they said: "As retpoline and RSB stuffing completely mitigate the Spectre Variant 2 issue for the Linux Kernel, SUSE decided, with guidance from Intel, to remove the IBRS patches from our kernel releases. While on Intel Skylake there exists a theoretical possibility that the software mitigations are not complete, so far no research has shown exploitable scenarios. Should research show any exploitable scenarios there, SUSE will reenable the IBRS mitigation on these chipsets."

So now that research has shown exploitable scenarios, I'm sure that's what they've been doing the last week. This means that the clever "no hardware required" retpoline hack that Google had originally invented to protect their Chromium browser from these attacks worked for about three years until enough time, focus, and reverse engineering had been applied by some very dedicated researchers to hack past the imperfect mitigation that retpoline was and turn a theoretical vulnerability into a very real threat.

Meanwhile, the day before yesterday, on Sunday, Linus posted into the Linux Kernel 5.19-rc7 thread, writing: "It's a Sunday afternoon. I wonder what that might mean." He said: "Another week, another rc. We obviously had that whole RetBleed thing, and it does show up in both the diffstat and the shortlog, and rc7 is definitely bigger than usual. And also as usual, when we've had one of those embargoed hardware issues pending" - meaning all of this RetBleed stuff - "the patches did not get the open development, and then as a result missed all of the usual sanity checking by all of the automation build and test infrastructure we have. So no surprise. There has been various small fix-up patches afterwards, too, for some corner cases.

"That said, last week there were two other development trees that independently also asked for an extension, so 5.19 will be one of those releases that have an additional rc8 next weekend before the final release. We had some last-minute btrfs reverts, and also there's a pending issue with an Intel GPU firmware." So anyway, this did affect the Linux kernel delaying its release by a week so that they could get the IBRS stuff back in and going.

Now, among all of this, more than anything else, I loved Intel's description of this problem. It's CVE-2022-29901. And it starts out with the phrase, "non-transparent sharing." Now, okay. You've got to love that. Somewhere in their technical press release department, someone called out: "Hey, anyone. I need a term for 'leakage' that doesn't sound like a bad thing." And someone replied: "How about non-transparent sharing?" The writer said "perfect," returned to his keyboard, and wrote: "Non-transparent sharing of branch predictor targets between contexts in some Intel processors may allow an authorized user" - meaning someone logged in - "to potentially enable information disclosure via local access."

Okay. Again, non-transparent sharing of branch predictor targets between contexts. Okay, anyway, everyone gets the message. They could not possibly have soft-pedaled this thing any more than they did. What it means is at least several hundred bytes per second can be exfiltrated from your Linux kernel if you don't fix this. Whoops.

Okay. The good news is, not all processors will be affected. The ETH Zurich researchers said they tested the RetBleed attack in practice on AMD Zen 1, Zen 1+, and Zen 2, as well as the Intel Core generations 6, 7, and 8. This essentially means Intel CPUs from between three and six years ago, and AMD processors from between one and 11 years ago will likely be affected.

Fortunately, the industry is getting better about addressing these sorts of problems, and patches for RetBleed were incorporated into this month's Patch Tuesday in both OS and cloud infrastructure updates from all major providers. So that. This leaves us, though, with the performance hit that comes whenever we disable some performance-enhancing bit that had inherently exploitable features. We've talked about this from the first glimmer of the first of these many microarchitectural side-channel vulnerabilities. Since all of these fancy features were invented to speed up the execution of real-world code, taking them out or shutting them down, like when we need the most, means some performance loss.

The ETH researchers noted that installing these patches will have an impact on the CPU's performance metrics on affected processors between 14% and 39%. And another issue they found in AMD processors only, that they named Phantom JMPs, might even come with a 209% performance overhead. Yikes.

The ETH researchers concluded their paper by writing: "We showed how return instructions can be hijacked to achieve arbitrary speculative code execution under certain microarchitecture-dependent conditions. We learned these conditions by reverse engineering the previously unknown details of indirect branch prediction on Intel and AMD microarchitectures and its interaction with the Return Stack Buffer. We found many vulnerable returns under these conditions, using a new dynamic analysis framework which we built on top of standard Linux kernel testing and debugging facilities.

"Furthermore, we showed that an unprivileged process can control the destination of these kernel returns by poisoning the Branch Target Buffer using invalid architectural page faults. Based on these insights, our end-to-end exploit, RetBleed, can leak arbitrary kernel data as an unprivileged process running on a system with the latest Linux kernel" - that is, until last Tuesday. Actually, that's a good question. Until this coming next release, probably - "with all deployed mitigations enabled. Our efforts," they said, "led to deployed mitigations against RetBleed" - oh. "Led to deployed mitigations against RetBleed in the Linux kernel." So presumably that has been resolved. So, yay.

And Jason, I look forward to you coming back for the next bleeding attack that the industry suffers.

JASON: I don't need this attached to me, this whole idea of when I return to the show something bleeds.

Steve: If we could get the name of Jason, I want JasonBleed. I think that'd be very cool.

JASON: No, I don't. I don't. I appreciate that we all want different things. I do not want that. But we'll see. Who knows? Maybe we can break the cycle, and there will be no bleeding on the next time that I return. Let's hope.

Steve: Let's hope. Let's hope.

JASON: Thank you, Steve, for sharing all of your wisdom on this and everything else throughout the show. You can find everything that Steve does at GRC.com. That's where you can go to find, well, everything you need to know about SpinRite, of course, that best mass storage recovery and maintenance tool. You can get your copy right there.

You can find audio and video of this show at GRC.com, also transcripts at GRC.com, which is I think the only place you can get transcripts of this show. I don't believe that we offer those on our site, but you can find them there.

If you want to go to our site, there is the show page on the web, TWiT.tv/sn for Security Now!. You can find audio, video. You can jump out to YouTube. Everything you need to know about the show is listed there, as well, including our recording times. We record live every Tuesday at 4:30 p.m. Eastern, 1:30 p.m. Pacific, that's 20:30 UTC. So if you want to watch live, you can do that, TWiT.tv/live. And you can follow along on all the bleeding each and every time that I'm joining Steve on Security Now!.

Steve, thank you so much for doing the show once again with me. I appreciate you welcoming me back.