Scanning the Internet

Description: This week we examine what happens to your monthly cloud services bill if you're infected by cryptomining malware. And speaking of cloud services, is Elon paying his bills? Just how fast are IoT-based DDoS attacks rising? What was the strange tale of wayward Chinese certificate authority? What useful new privacy and security features will Apple be adding to their services with their net OSes this fall? And why has France headed in another direction? How does Russia feel about foreign Internet probes and what can they do about it? And after a bit of miscellany, listener feedback, and a SpinRite update, we're going to take a deep dive into the back story and current capabilities of the Internet's premier scanning and indexing service, Censys.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-927.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-927-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with a big, big, big show for you coming up in just a little bit. Cryptomining, who pays for all that cryptomining? Who doesn't pay for their Google Cloud services? A Chinese certificate authority that you really shouldn't be using if you can avoid it. And then Steve looks at Apple's security announcements from last week's WWDC. That and a whole lot more, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 927, recorded Tuesday, June 13th, 2023: Scanning the Internet.

It's time for Security Now!, the hatless Steve Gibson. But you don't need a hat because he's inside to do the show. Steve Gibson's here, our master of ceremonies and all of that. Hi there.

Steve Gibson: Hello, Leo.

Leo: Hey there.

Steve: And as we know, glass is a blocker of UVB radiation, which is why we should all be taking some Vitamin D because we're not getting it when we're inside. And that's why when you hang your arm out the window, when the windows rolled down in the old days...

Leo: Get the farmer's tan.

Steve: Yeah, that's right. You've got the arm sunburned and nothing else.

Leo: I know that, because I have those new transitions lenses on my eyeglasses, that UVB darkens them. But you're sitting in the car, and they don't darken, which is kind of a problem. Anyway, we're not here to discuss optics. We're here to discuss security. What's up?

Steve: We are. We're going to - I think this is going to be a fun one, not that that's any surprise, I hope. We're going to examine what happens to monthly cloud services billing if you get infected by a cryptomining malware. It's not good, but Google has an answer. And speaking of cloud services, is Elon paying his bills? And what's about to happen? Just how fast are IoT-based DDoS attacks on the rise? What was the strange tale of a wayward Chinese certificate authority, and why it's good what happened? What useful new privacy and security features will Apple be adding to their services with their next OSes in the fall?

We've got an after-WWDC report from last week's event. And why has France headed in another direction from Apple? How does Russia feel about foreign Internet probes entering their IP space? And what, if anything, can they do about it? And after a bit of miscellany, some listener feedback - one of which drives me into a brief sci-fi reading retrospective for listeners, we haven't talked about that for a long time - and then a quick SpinRite update, we're going to take a deep dive into the back story and the current capabilities of the Internet's premier scanning and indexing service, which is known as Censys (C-E-N-S-Y-S). Thus today's title is "Scanning the Internet."

Leo: Oh, Censys with a Y.

Steve: Censys with a Y.

Leo: That's because there's no U in it.

Steve: That's right. And as they say, there's no U in Internet.

Leo: That's right.

Steve: It's not about you. It's about the world.

Leo: Me. All right.

Steve: That's right.

Leo: Time for the Picture of the Week, Steve.

Steve: So this is sort of astonishing. And maybe it's obvious in retrospect. But I have to salute the unnamed individual who thought, you know, clearly somebody who was very much into the generative AI image stuff, like could I use Stable Diffusion to morph a real-world photo, an image, into something that a QR code reader would perceive as a valid QR code? And the answer was yes. This thing tore through Reddit last week. People were just blown away by it.

So our Picture of the Week shows an outdoor sort of a vegetable market with some stands and various shoppers and purveyors, some sort of underneath some sun protective awnings. And it is also a valid QR code. I mean, it looks maybe a little odd, but surprisingly not. And but then you notice the three standard QR code targets, two on either upper corner and one in the lower left, where if you sort of get your eye to look at it right, you can see that. And then like down on the sidewalk, down toward the bottom of the picture, it sort of looks like shadowing, but it's actually part of the QR code. And the more you look around, the more you can sort of see how the Stable Diffusion engine fit the needs of the QR code into the image. And these were so cool that I have a second page of them in the show notes, showing just two more examples, or actually four more examples, of images that were morphed using Stable Diffusion to create what's arguably beautiful QR codes.

Anyway, so in case anyone was interested, I've got a link in the show notes: stable-diffusion-art.com/qr-code. And at the top of the page it said: "A recent Reddit post showcased a series of artistic QR codes created with Stable Diffusion. Those QR codes were generated with a custom-trained ControlNet model. Just like another day in the Stable Diffusion community, people have quickly figured out how to make QR codes with Stable Diffusion without a custom model." And then this page goes on to show you how you can indeed create your own. Give it an image you want and a QR code you have, and presto, here's like this amazing hybrid. So anyway, just I thought it was very cool and wanted to share it with our listeners.

Okay. So what happens, more often than not, when a cloud computing account is compromised? Turns out the bad guys waste little time setting up and running a cryptocurrency mining operation. The bad news for unwitting users is that, as we know, the reason this is being done is to mine on someone else's dime. And the more computational resources that are available, the greater the rate of currency minting. Consequently, minting on stolen accounts is typically not throttled, and it can consume massive amounts of compute time in a short bit of real time.

And that brings us to the question: Who pays for that stolen compute resource usage? Well, Google's June 8th announcement was titled "New Cryptomining Protection Program offers $1 million for costly cryptomining attacks." It's like, yes, I mean, you could get hit with a bill that just astonishes you at the end of the month. And it's funny because what I remember was like in the early days of the Internet, there was this notion, it was called 90 - well, actually it still exists when you're dealing with top-tier providers, 95-5 Internet bandwidth billing.

The idea was that the ISP wanted to make available higher bandwidth in bursts, but didn't want to get taken advantage of. So over the course of a month all of the - essentially the amount of bandwidth that a customer was using would be sampled in small pieces, then all of those samples would be sorted from greatest to least amount of bandwidth used in that sample. And then the 95th percentile was taken. Maybe I've got it sorted backwards. I think it was sort of the other direction. Anyway, the idea was that if you had a - if you were using high periods of bandwidth, which wasn't like all month, it wasn't average bandwidth, it was weighted so that if you crossed more than 5% of the samples, the amount of bandwidth used at that 5%-95% percentile point, that was then taken to be your bandwidth usage for the entire month.

So the point is that, if you weren't careful, you could similarly get hit with an astonishing bandwidth charge from an ISP that was billing in this way. So anyway, this is sort of the same. It's like, okay, you're not paying attention. A bot crawled onto, you know, got into your cloud computing account and has spent like all month grinding away, generating cryptocurrency, and then Google says "You owe us $5 million." It's like, what?

Anyway, so their announcement reads more like a promotional advertisement, but it contains some useful information so I wanted to share it. Here's what they said. They said: "Cryptomining is a pervasive and costly threat to cloud environments. A single attack can result in unauthorized compute costs of hundreds of thousands of dollars in just days. Furthermore, the September 2022 Threat Horizons Report published by Google's Cybersecurity Action Team revealed that 65%" - okay, so just shy of two out of every three - "compromised cloud accounts experienced cryptocurrency mining." So as I said, if they can get into your cloud account, that's what they do.

Google said: "Stopping a cryptomining attack requires effective detection, which is why we've made it a focus of Security Command Center Premium, our built-in security and risk management solution for Google Cloud. To strengthen our customers' confidence in their ability to quickly detect and stop cryptomining attacks" - basically they're going to give us indemnification. They said: "We are introducing a new Cryptomining Protection Program which offers financial protection up to $1 million to cover unauthorized Google Cloud compute expenses associated with undetected cryptomining attacks for Security Command Center Premium customers." In other words, basically they're saying sign up for this, pay for this, and we'll indemnify you because we're sure we're going to be able to detect when this is happening on your account.

So they said: "We're able to offer financial protection because Security Command Center Premium includes specialized detection capabilities that are engineered into the Google Cloud infrastructure. To detect cryptomining attacks, Security Command Center scans virtual machine memory for malware. It does this without agents, which can slow performance and increase an organization's attack surface. Our approach enables us to detect attacks that could be missed by bolt-on security tools that rely on analysis of cloud logs and information gathered from APIs.

"Security Command Center can also detect compromised identities, which allow attackers to gain unauthorized access to cloud accounts and quickly deploy cryptomining malware. This means Security Command Center can detect possible threats before an adversary can exploit compromised information to begin an attack. This full set of advanced detection capabilities for cryptomining can only be delivered by a product built into the cloud infrastructure," blah blah blah. So, yeah. A commercial service, but just sort of brings to mind, you know, this is the kind of thing, unless you're really watching your cloud system closely, or really do have your own technology for detecting when suddenly your CPU usage gets pinned, and the meter starts running fast at Google, this would make sense, I think.

They quoted a guy named Philip Bues, who's IDC's research manager for cloud security. Of course it's pro Google. But he said: "Cryptomining attacks" - this is IDC. "Cryptomining attacks continue to be a serious security and financial issue for organizations who do not have the right preventive controls and threat detection capabilities in their cloud environment." So just don't go setting it up and forgetting about it and assuming everything's going to be fine.

He said: "Google Cloud is taking an important step by providing built-in threat detection of unauthorized cryptomining, backed by real financial protection available to Security Command Center Premium customers. If an attacker evades their detection defenses, they're there to back you up. This shared fate approach to cloud security helps increase confidence among enterprise buyers when moving to the cloud." And of course this is

also, you know, this helps Google; right? Because they've got, in their whole infrastructure, everybody is sharing this pooled compute resource. So they don't want it to be drained off by a bunch of undetected malware which is reducing the total amount of compute that is available to everybody who's paying for the cloud services.

Leo: They may not mind if you pay for it. But still, it's probably a good thing. You've got to wonder, I mean, this is kind of a shame that crypto exists at this point. I feel like it's the negatives of crypto have really outweighed any benefits.

Steve: You know, I hadn't thought that, Leo, but you're absolutely right. We talked about it, gosh, years ago.

Leo: We thought it was interesting, you know, it was a really cool, you know, Satoshi's original paper and all of that. But as it turns out it just really enabled ransomware and this kind of cryptomining hijacking.

Steve: And all the speculation. People are losing their shirts. In fact, it didn't make it into the show notes this week, but the FCC...

Leo: Coinbase and Binance have both been sued by the FCC.

Steve: Yes, yes.

Leo: For unregulated securities exchanges.

Steve: Yes. But also the very first exchange that fell, Mt. Gox, it turns out that $1.7-some billion of current value in Bitcoin was slowly siphoned out of Mt. Gox over years before it finally went under. So it's interesting. I hadn't thought of that, Leo, but you're absolutely right. I agree.

Leo: Not to mention the environmental impact of these servers cranking up at high speed.

Steve: Yeah, Niagara Falls is much warmer now than it used to be.

Leo: There's not going to be any ice this summer in the Arctic, and I think you can blame Bitcoin. I plan to, anyway. Geez.

Steve: Well, speaking of blaming people, while we're on the topic of Google Cloud Services billing, and even though this is a bit more gossipy than our usual fare, but since it is potentially an intriguing event in our industry, I decided to share the news that Elon Musk's Twitter has reportedly been refusing to pay both its Google Cloud and Amazon AWS bills, in an apparent strong-arm play to renegotiate its preexisting multiyear contracts which Twitter had signed with both service providers. And this has been going on long enough now to lead both companies to independently begin threatening

termination of services. In the case of Google Cloud, this reportedly leaves Twitter's trust and safety systems, which maybe Elon doesn't care that much about, hanging in the balance as Twitter's contract with Google Cloud Services comes up for renewal this month. And we're about mid-month at this point.

Although Twitter hosts some services on its own servers, the company has long contracted with both Google and Amazon to complement its infrastructure. And prior to Musk's acquisition of Twitter last year, Twitter had signed an extensive multiyear contract with Google to host services related to, among other things, fighting spam, removing child sexual abuse material, and protecting its users' accounts. So those facets of Twitter's services might be, well, you know, we'd like to have them; but, you know, Google wanted us to pay. So, sorry.

Anyway, after acquiring Twitter, Musk reportedly issued a blanket mandate to his minions, requiring them to cut $1 billion from Twitter's infrastructure costs. So it may be that he feels that playing hardball with Google and Amazon is the way to at least begin the process of renegotiating those agreements which predated his acquisition. You know, we've previously heard stories about Twitter choosing to default on its existing physical office lease agreements.

Leo: Rent, yeah, yeah.

Steve: And if Twitter manages to cut a billion dollars from its infrastructure costs, as I said, it may be eliminating the cost of various protective services it's able to provide to its user community.

Leo: Yeah, well, who needs those?

Steve: Wow. Yeah, you know.

Leo: Needless expense.

Steve: Kiddie porn, well, what are you going to do? You know, we can't look at everybody's feed. Over on the Amazon side, since Twitter has also been delaying its payment for Amazon's Web Services, Amazon has reportedly been threatening now to withhold its advertising payments to Twitter, which would, of course, impact Twitter's revenue. So maybe that'll get Elon's attention. Anyway, it all seems like a big mess. But, you know, messes seem to follow Elon around. So we'll see what happens. Anyway, I just thought - I ran across it, I thought, well, okay, we're on the topic of paying for cloud services. So Elon says that we don't have to.

As our listeners know, the risks posed by the rapid uptake and proliferation of today's not-yet-really-secure IoT devices has been a constant source and topic of concern here. Last Wednesday, June 7th, a report published by Nokia's Threat Intelligence team gave these concerns some numbers. Here's what Nokia's report explained.

They said: "The latest Nokia Threat Intelligence Report released today" - which was last Wednesday - "has found that IoT botnet DDoS traffic, originating from a large number of insecure IoT devices with the aim of disrupting network services for millions of users, increased fivefold over the past year, following Russia's invasion of Ukraine and

stemming from the growing increase in profit-driven hacking collectives operated by cybercriminals." And Leo, yes, paid for by cryptocurrency.

Leo: Uh-huh.

Steve: In cryptocurrency. "This sharp increase, also supplemented by the increased use of IoT devices by consumers around the world, was first noticed at the beginning of the Russia-Ukraine conflict, but has since spread to other parts of the world, with botnet-driven DDoS attacks being used to disrupt networks as well as other critical infrastructure and services. The number of IoT devices (bots) engaged in botnet-driven DDoS attacks rose from around" - okay, so this is the number of IoT devices - "rose from around 200,000 a year ago to approximately one million devices, generating more than 40% of all DDoS traffic today."

So once it was that you were commandeering big iron servers where you could because they had big pipes connecting them to the Internet, and lots of network juice, and you could pump out a lot of packets. Now we're distributing. We've got a million IoT devices, each on residential Internet connectivity. But, boy, you take the typical residential Internet and multiply that by a million, and, well, we've been talking about the size of recent DDoS attacks. They're just astonishing in size.

So Nokia said: "The most common malware was found to be a botnet malware that scans for other vulnerable devices, a tactic associated with a variety of IoT botnets. There are billions of IoT devices worldwide, ranging from smart refrigerators, medical sensors, and smart watches, many of which have lax security protections. The Threat Intelligence Report also found that the number of trojans targeting personal banking information in mobile devices has doubled to 9%, putting millions of users around the world at heightened risk of having their personal financial and credit card information stolen.

"The report, however, did find some encouraging news, showing that malware infections in home networks declined from a Covid-high of 3%" - but still, 3% of home networks infected. It was cut in half to 1.5%.

Leo: Wow.

Steve: And they said - still, yeah: "Close to the pre-pandemic level of 1%." So it hasn't even dropped back down.

Leo: That's a good job done by Windows, I think, primarily, by Microsoft.

Steve: I would think that's probably the case, yes, yes.

Leo: Yeah.

Steve: They said: "As malware campaigns targeting the wave of at-home workers tapered off, and more people returned to office work environments. These findings are based on data aggregated from monitoring network traffic on more than 200 million devices globally" - that's Nokia's view into the Internet - "where Nokia NetGuard Endpoint Security product is deployed."

Nokia's Senior Vice President for Business Applications said: "The key findings in this report underline both the scale and sophistication of cybercriminal activity today." And of course that's something we've been noting here constantly. This is, you know, the world has changed. It was at the beginning of this podcast, it was oh, look, I wrote a worm; you know? Now it's, okay, like there's now a darknet which is saying, okay, we've got 200,000 IoT bots in our net. Pay us, and we'll beam there wherever you tell us to. It's a completely different complexion today.

He says: "A single botnet DDoS attack can involve hundreds of thousands of IoT devices representing a significant threat to networks globally. To mitigate the risks, it is essential that service providers, vendors, and regulators work to develop more robust network security measures, including implementing threat detection and response, as well as robust security practices and awareness at all company levels." And, you know, we've often observed, Leo, that a lot of this is spoofing IPs. Unfortunately, when you've got a million individual IoT devices, they don't even need to spoof their IP. No one cares if they get identified. It's some light switch, or some plug.

Leo: Right.

Steve: Just like there's too many of them to deal with. You just have to do what Cloudflare does and bring up transient defenses in order to block connections at the perimeter and prevent them from getting through.

Leo: The darker interpretation of this is not that our computers are more secure, but the malware creators are going to the softer targets in IoT. And there's no softer target than IoT.

Steve: Right. They don't care if it's a toothbrush that's generating IoT traffic or a mainframe. To them all they are is packets, packet missiles that they're able to launch at a temporary enemy.

Leo: I suspect that really that's the reason is just these targets are easier to compromise.

Steve: Well, and now, why is it happening? They're getting paid.

Leo: Oh, there's money in it. Oh, yeah.

Steve: Yes. It's now a business. It's a business. Remember in the beginning the original botnets were script kiddies who were blasting their competitors off IRC servers.

Leo: Yeah. Have you read the new, or are you going to read the new book about Fancy Bear, or it's called "Fancy Bear"? It's really a good book ["Fancy Bear Goes Phishing"]. I recommend it.

Steve: Oh, okay.

Leo: It starts with the Robert Morris worm. But he takes five well-known malware attacks, Fancy Bear is one of them, and talks about it. His final conclusion is interesting. Because these are Turing machines, they cannot be secure. It's just the nature of it. They can do anything, including malware, and they always will be. But I didn't, you know, pro or con on that, the stories themselves are great. And Fancy Bear, to your point, which eventually became the Mirai Botnet and all of that, the kid who did Mirai - it wasn't Fancy Bear. It was the kid. Fancy Bear's the Russian group; right?

Steve: Right.

Leo: The kid who did Mirai was a Rutgers students in his freshman year. He was pissed off that Rutgers wouldn't allow underclassmen to take the computer science courses he wanted, so he wrote his first DDoS attack to take the Rutgers network down so nobody could register for any classes. But to your point, and then later became more malicious, and he did Mirai, and eventually got caught by the FBI. It's a really good story. That's been excerpted. You can read it around if you want.

Steve: Cool, sounds great. Yeah, really.

Leo: The book is called "Fancy Bear." That's why I was thinking of Fancy Bear.

Steve: Right, right. That's a good title for a book.

Leo: Yes, it is.

Steve: It's really much more catchy than APT28.

Leo: Yeah. Yeah, I agree.

Steve: And that's not going to jump off the shelves.

Leo: Or the GRU or whatever. Yeah, I agree, 100%.

Steve: Okay. So a guy named Matt Holt wrote a nice little web server in the Go language. He calls it Caddy Server, you know, as in "Caddy Shack," Caddy Server, and describes it as an extensible, cross-platform, open-source web server written in Go. The name "Caddy" refers to both a helper for tedious tasks, like someone carrying someone's golf clubs, and a way to organize multiple parts into a simplified system.

Okay. So we've established that Matt knows his way around web server technology. He was experimenting with ACME, which is, as we know, the protocol created by the EFF's Let's Encrypt project to automate the issuance of TLS certificates, the idea being that it allows the server to proactively ask a certificate authority to please refresh its cert because under the Let's Encrypt model, certificates only last 90 days. So you want to automate that process so that you're not having to do it constantly.

Okay. A low-budget Chinese certificate authority named HiCA (H-i-C-A) only supported one particular ACME client.

Leo: Oh, that's suspicious.

Steve: Uh-huh, for its customers' servers; right? Because if it's an open protocol, and it is, it should not matter. Matt found this odd, just as you did, Leo. Now, this client, this one particular ACME client, is open source, and it's over on GitHub as ACME.sh. So here's a bit of what Matt wrote. He said: "HiCA's documentation" - that is, the documentation with this certificate authority - "explains that it only supports acme.sh as a client. This was curious to me, so I tried to learn why. If it's using ACME" - and this is the thing that really got him, the ACME logo - "it should be basically compatible with the majority of ACME clients. While observing a certificate using ACMEz, I discovered that the discovery was blocked" - and this is like details of the protocol we won't get into because you'll get the gist of this - "the discovery was blocked unless the User-Agent is set to a string that starts with Mozilla or acme.sh/2.8.2.

"Once I faked the User-Agent in my own client and got that working, certificate issuance still failed. Curiously, the error message involved trying a URL of dot dot" - meaning, you know, backup in a hierarchy - "../pki-validation. This doesn't make any sense to me, even though that kind of appears in their docs, because it's not standard ACME. So I dug a little deeper to figure out what the Challenge object consisted of that would cause my client to be making a request to ../pki-validation.

"It turns out that the Challenge object looks unusual, and it became immediately obvious to me why HiCA only supports acme.sh. They are not conforming to ACME at all." And he says: "(Bugs the heck out of me that they're using the official ACME logo on their site, even though they don't implement the ACME standard.)" And he says: "Instead, HiCA" - you sitting down, Leo? "HiCA is stealthily crafting curl commands and piping the output to bash. Acme.sh is being tricked into..."

Leo: What could possibly go wrong?

Steve: Oh. "Acme.sh is being tricked into running arbitrary code on the remote server." Okay. So let me make that a bit more clear and fill in some additional details. A small Chinese certificate authority requires their clients, their users, their customers, to only run a specific acme.sh ACME client, specifically because this particular open source client has a bug which the CA has been exploiting to cause their clients' web servers, their customers' web servers, to remotely execute arbitrary code and commands on their own servers.

Leo: Oh, my god.

Steve: Wow. Now, obviously, no one should ever run code, meaning ACME clients, that they don't trust on their servers. And if some certificate authority tells you that they support ACME, but only one specific ACME client, even if their certificates are free, run away as fast as you can.

Leo: So this is probably the Chinese Communist Party, the Chinese government.

Steve: Oh, Leo, it's a plot. They're trying to take over our children.

Leo: No. But they're hoping that some foolish industrial enterprise will use the server and so they can keep an eye out; right? I mean, I don't think it's hackers. It's probably the government. But who knows; right?

Steve: No. Well, based on the - I read the GitHub thread discussion. The guy behind this HiCA got involved when this became a controversy. And he appears to be benign and goodhearted. And of course if it was actually evil Commies, then he would. He explained that doing this allowed him to have more flexibility.

Leo: Oh, yeah.

Steve: And if he's able to run whatever code that he wants on your server, that's flexible.

Leo: Could be benign. It could really be just some [crosstalk].

Steve: Yes, yes. I think it was just he saw a bug, and he decided to use it. The acme.sh maintainers immediately fixed the bug that this HiCA guy was exploiting for their service, and HiCA shut down and closed its doors. It wasn't a big deal. He was affiliated with some other CA. It was not quite free, but it was like $3.12 or something. So he wasn't making a lot of money. And he said, okay, fine, I'll just stop this because this isn't fun anymore.

Leo: Yeah.

Steve: But still, you know, everybody who was using his certs was having his code running on their computer.

Leo: Jiminy Christmas. So it wasn't the server, it was the certificate authority. Interesting.

Steve: Yes. The certificate authority has a...

Leo: He found a bug. I get it. He found a bug in the server that he was able to exploit. I get it now.

Steve: Yeah. He found a bug in the acme.sh client.

Leo: Right.

Steve: So his customers had to run, by his instruction, the acme.sh client in order to get the certs from him. And the way they were getting them from him is he was saying, okay, you want a new update? I need to run a little code in your server in order to make that happen. What could possibly go wrong?

Leo: And we never saw a malicious attack.

Steve: No, no.

Leo: So maybe it was completely benign.

Steve: Right. I think so. But again, this is the way we keep these things from escalating.

Leo: Yes, exactly. And bravo to what's his name, Matt Holt, for finding this and writing it up. And use Let's Encrypt, please. Just use Let's Encrypt.

Steve: Exactly. Why do anything else? Use the one everybody uses, that's been vetted and had real security people writing it, maintaining it. I don't get the need for additional ACME clients. But again, everything's open, and it should be. But it does not prevent bad ones from being written. And, you know, I see some cars, Leo, that have been painted a color which should not have been allowed to leave, like, the garage.

Leo: Cars, forget it. We've got Victorian houses painted that way up here. Fluorescent.

Steve: So, yeah. If they build it, somebody will...

Leo: Somebody will use it, yeah. That's right. So what a world.

Steve: Okay. So Apple's 2023 World Wide Developer Conference is now behind us, and it was very interesting. Apple did not disappoint with their continuing focus upon the privacy and security of their users. It's very clear that they intend to offer both privacy and security as features of their products and technologies. For example, during the presentation of their new mixed-reality vision goggle system, they made a point of noting that the system's quite powerful eye-tracking technology creates an inherent privacy risk, which at first you think, huh? But social scientists have long understood, where a user's eyes look when confronted with an image reveals, with surprising fidelity, the innate emotional power of the content of various parts of an overall image.

Now, interpreting what that means exactly may be problematic, but it's still an unintended gateway into a user's mind. So when we ask the question, do you want a web page you visit to know where you looked on that page, less privacy-centric developers might think that would be quite cool, and might sell it as a feature.

Leo: Oh, yeah. We do heat maps to see what people are looking at. Yeah, it's a usage thing, to find out how they use the site, yeah.

Steve: Exactly, like having a virtual mouse pointer automatically jump to that location where then a page's JavaScript is able to obtain its coordinates and relay them back to the mothership. But that's not the way Apple thinks. Apple was quite clear in their presentation that where a user's eyes were looking was private information that would never leave that device.

Leo: Good. Good.

Steve: Yes. And again, this is Apple on our side. Only when they looked and clicked, which means did a little finger pinch thing, would the location of that click be returned, just like clicking the mouse pointer on something. So it's this pervasive attitude across Apple that led me last week to opine that there's no way Apple has deliberately supplied anyone, including our own NSA, with a robust backdoor through iMessage to launch iDevice malware. That just isn't something they're going to do.

Okay. So back to last week's WWDC. Although this year's advancements did not explicitly focus upon user security nearly to the degree that they did last year, that was a big focus of WWDC 2022, Apple still demonstrated that this continues to be a selling point for them, and they're going to sell it. So for their Safari browser, Apple says that they've added additional tracking and fingerprinting protections which go even further to help prevent websites from using the latest techniques to track and identify a user's device. This is the constant cat-and-mouse game, and they're just continuing to tighten it up.

Also, Safari's Private Browsing mode now locks when it's not in use to allow a user to leave private tabs open even when they've stepped away from the device. Safari will now show a locked browsing window and request a Touch ID or password or Face ID in order to unlock and view those tabs. And Safari's Private Browsing windows now automatically lock as a whole, if they've not been in use recently. So nice moves there.

In Photos, a new embedded photos picker can help users share specific photos with apps while keeping the rest of their library private. So they've made it more granular. When apps ask to access the user's entire photo library, the user will be shown more information about what they'll be sharing, along with occasional reminders of their choice. And I think that is so important. This notion of occasional reminders that previous permissions remain in effect represents a significant advancement in our understanding of the human factors side of how to offer security and privacy. It's so easy for us to grant a permission in the moment when we want to make something specific happen, but then to leave that permission enabled well after it's no longer appropriate. So a gentle nudge to ask, "Huh, is this still want you want?" That's just brilliant privacy-enhancing tactics.

So in the case of Photos, the Photos permission prompt now tells users how many photos and videos they would be giving access to, as well as providing a sample of those photos. Apple is also moving to curtail the surreptitious link tracking which occurs in Messages, Mail, and Safari's Private Browsing. It's becoming commonplace, more commonplace, for websites to append extra information onto to their URLs as a means of tracking users across sites.

We've talked about this years ago, how the "Referer" header informs advertisers of the URL of the page which is pulling the ad. If this URL is needlessly embellished with enhanced tracking info, that information gets sent. And there's been no way to automatically limit this. Apple says that they're changing this, or will be this fall in iOS 17, by silently removing this unnecessary information from the links users share in Messages and Mail, and from the links in Safari's Private Browsing.

And what Apple calls Communications Safety is also being further advanced. Communication Safety, which has been designed to warn children when receiving or sending photos in Messages that contain nudity, now also covers video content in addition to still images. And a new API lets developers integrate this Communication Safety into their own apps. So this would allow these warnings to be present in non-Apple apps, as well.

And this Communication Safety with now also help keep kids safe when they're sending and receiving an AirDrop, a FaceTime video message, and when using the Phone app to receive a Contact Poster and the Photos picker to choose content to send. So basically they launched this under 16, verified that it's working, and now they're confident enough to extend this out into additional apps that could be problematic.

All image and video processing for Communication Safety occurs only on the device, so that neither Apple nor any third party gets access to the content. And as we've talked about before, these warnings will be turned on for the child accounts in their Family Sharing plan, which can be disabled by the parent. A Sensitive Content Warning is shown in Messages in the iPad Pro.

What Apple calls Communication Safety is what protects children receiving or attempting to send videos or photos, and the same protections are available for adult users in the form of a Sensitive Content Warning. The feature is optional and can be turned on by the user in Privacy & Security settings. And as with Communication Safety, all image and video processing for Sensitive Content Warnings for adults occurs on the device, meaning that nobody else gets to see it.

And there was some conversation of this, as well, I think I saw it on some of the podcasts, Leo, probably on MacBreak Weekly. Apple has also added Passwords and Passkey sharing with the creation of sharing groups, which totally makes sense to do. Users can now create a group to share a set of passwords, and everyone in the group can add and edit passwords to keep them up to date, as needed. And in a slick new feature that I want to see in action, this will be really interesting to see. Apple says that one-time verification codes received in Mail will now automatically autofill in Safari without the user leaving the browser.

Okay. So okay. It sounds as though you're on a web page that sends you a link, where the web page wants you to authenticate your email address by giving it a six-digit code or whatever. So it says we sent you email, please populate this field with your code. Apparently iOS will be observing that empty and waiting field and also notice that you've just received email in the background which contains a code. So it will parse the email for the code and populate the one-time code field in the browser so that it just appears, and then you just click on, yeah, there's your code. And it happens. So wow, I can't wait to see that happen, you know, in front of me. That'll be very cool.

We've talked about Apple's "Lockdown Mode" which significantly reduces the iPhone attack surface by dramatically restricting the content that the phone will accept and process. Apple is pushing this technology now even further. This was also the first time I've seen the term "mercenary spyware" anywhere. I love the term. I'm going to be using it. These new lockdown protections encompass safer wireless connectivity defaults, media handling, media sharing defaults, sandboxing, and network security optimizations.

So now, when we get this in iOS 17, enabling Lockdown Mode will further harden device defenses and strictly limit functionality, all in the name of security. And I think it makes a lot of sense. Oh, and it's also coming to watchOS. When you enable Lockdown Mode on your iPhone that is paired with your watch, the watch also gets locked down. So, you know, it's not like they've left a way in by sneaking in through your watch.

Also, Apple has something that they're calling "Check In," which is an interesting new feature. I'll just share how Apple described it. They said: "Check In makes it easy for users to let friends or family members know they've reached their destination safely. Once turned on by the user, Check In automatically detects when the user has reached their intended destination, and will let selected contacts know via Messages. In the case that something unexpected happens when the user is on their way, Check In will recognize that the user is not making progress toward their declared destination and check in with them. If they don't respond, the feature will share useful information like the user's precise location, battery level, cell service status, and the last active time that they used their phone with the contacts and the users selected.

"In addition to making it easier to get help if needed, Check In is designed around privacy and security" - of course, it's Apple - "keeping the user in control by letting them choose with whom to share their information, including the destination and time duration that they set. Users can end the Check In session at any time," you know, canceling it. "Information sent with Check In is end-to-end encrypted so only the user's family member or friends who have been authorized are able to read it, not Apple or anyone else." So again, just kind of somebody at Apple saying, what new service, how else could we help people leverage the technology that they're now carrying in their pocket? And so they created another one.

"NameDrop" is another new feature which allows for tightly controlled contact information sharing from one device to another, presumably enabled through NFC since the devices basically need to be in super close proximity, if not touching.

And, finally, a brilliant innovation which they call "Live Voicemail" allows the recipient of a phone call that they have chosen to let go to voicemail observe a real-time textual transcript of the voicemail as it's being recorded on their phone, and then change their mind on the fly to pick up the call. It's the brilliant modern equivalent of how we used to use residential telephone answering machines to screen calls, and then we'd grab the phone receiver to pick it up, you know, claiming that we had just walked in the door while they were leaving the message and heard them leaving it, after we heard who it was or what the call was about. So now we have the same thing on our smartphones.

All these goodies will be arriving later this year, presumably with iOS 17. And, you know, big props to Apple for continuing to raise the bar on security and privacy and, I think, doing everything they can for their users, and clearly selling these features as part of their product offerings.

Now, not exactly following Apple's example, we have France, who last Wednesday evening, June 7th, the French Senate passed an amendment to its so-called "Keeper of the Seals" justice bill. And I have no idea where that name came from. The approved changes, which passed last Wednesday, now allow law enforcement agencies to secretly activate the cameras and microphones of remote devices, and specifically smartphones, without notifying the device's owner.

Officials say they plan to use this new provision to capture sound and images of suspects of certain types of crimes. The measure would be reserved for cases of delinquency, organized crime, and terrorism. Delinquency? Okay. The same update to the bill text would also allow law enforcement agencies easier access to geolocation data to track criminals suspected of committing offenses punishable by at least 10 years in prison.

What's not mentioned is exactly how they intend to make this actually happen in practice. I had Google translate the French news web page, and they were saying that without this provision, investigators would need to plant physical bugs on the premises of their investigation targets. So this was being sold as a safer means for their investigators to accomplish the same already legal surveillance, like if you get a warrant to plant a

bug, by instead targeting their targets' phones, and turning the phones and other devices into surveillance equipment.

Now, we know that Apple's iPhones will actively resist any such abuse. There's no way to ask an iPhone to do that. But one wonders whether this might be paving a legal framework for the use of, to use Apple's new term, "mercenary spyware" such as Pegasus, which would subvert smartphone protections and would then, within the bounds of this legislation, no longer represent illegal spying which the country needs to deny and be ashamed of. So France did this last Wednesday evening.

Leo: Hmmm.

Steve: Yeah. Okay. So it should come as no surprise to anyone that Russia has decided to begin blocking foreign vulnerability scanning at the incoming border of RuNet. Very much like their continuing use of Microsoft Windows, my reaction to that is, "You're only getting around to doing that now? Really?" So, you know, there are services like Shodan and the one we're going to talk about at the end of the show, Censys, typical services.

There's also other security companies and proprietary scanners which are more or less, we know this, constantly poking around the entire Internet to see what they can find. When some security firm notes that, for example, some new vulnerability in a Cisco device affects more than 34,000 of them, well, that number comes as a result of scanning. That's how we know there's 34,000 of them out there hanging out on the Internet just waiting to hopefully get updates.

So it's entirely reasonable for an increasingly hostile foreign nation, like hostile to the rest of the world, not to want anyone poking around in their backyard. And wouldn't you know it, the responsibility for limiting such scans falls to our favorite Russian Internet watchdog, Roskomnadzor. In their announcement of this plan, they stated that more than 10 such services are constantly scanning inside their Russian RuNet for vulnerable systems that are then exploited in cyberattacks. And that number of scanners, 10, okay, that sounds about right.

The trouble is, to at least some degree, the scanners you know about are not the scanners you need to worry about. Shodan and Censys operate above board and scan from publicly known blocks of IP space. So blocking them, if one chose to, would not be difficult.

But as anyone knows who's ever tried logging all of the individual IP packet traffic arriving at any arbitrary IPv4 address, today there's more or less a continual flux of incoming noise. And as our long-term listeners know, long ago I coined the term "Internet Background Radiation" to remind us of exactly that. My point is, all of the IP space of Russia's RuNet is also constantly receiving this random noise flux. And it doesn't make sense to block it all, even if you could.

There's no way, for example, for any central authority to know which traffic to which port is part of the services legitimately being offered there and for whom those people want those incoming packets. Look at what a mess, for example, some cable providers make when they decide to block some ports that they don't think their subscribers should be using. It's not always good. So my point is, if some of those random-seeming packets were actually carefully aimed NSA probes, Russia would never be the wiser. The packets they do need to worry about would never be the ones belonging to the well-known public scanning services, which they've now said, okay, we don't want you anymore, and so they'll be blocked.

Okay. We've got some, well, actually one piece of Miscellany and some Closing the Loops. Tavis Ormandy, whom we've quoted through the years, he's @taviso, Tavis Ormandy at Google. On Friday, June 9th, last Friday, he tweeted: "Quick personal update. It's nearly 10 years since @scarybeasts and I started Project Zero. A lot has changed since then, and I've decided there are teams where I can have a bigger impact. I'm still at Google, and still working on vulnerability research. I'm going to work on CPU security with Google ISE." That's the Independent Security Evaluators. "We've already got" - and he's got like three flame emojis. "We've already got zero-day reports on the way."

So we're going to be seeing some interesting, and especially interesting if they're CPU zero-day reports. You know, that's different than Spectre and Meltdown, which are, you know, this might not be good someday. These are, uh, we just hacked your CPU. So stay tuned. Tavis moved, but he's certainly not gone.

Dave Johnston sent: "Hey, Steve. Big fan of the show. I heard your stat in Episode 926" - so that was last week - "about school districts lacking security staff. Having worked in K-12 and community colleges, I have some background. Many school districts are small, i.e., one or two schools. They might have one or two technology staff running the whole show. My last job was IT Director for a district with 7,000 students, 14 schools, which had two desktop techs, server admin, network admin, database admin, secretary, and a director."

He says: "That's not an unusual load. It's not that the districts don't care about cybersecurity. They're having a tough time just keeping all the technology running on a daily basis." And I can well imagine that. So David, thank you for the viewpoint from the trenches, from the front lines.

Mrlinux11, he said: "Getting error going to grc.sc/926." And this was sort of interesting because it was in a tweet, and there was a link. I clicked the link, and I got the error, too. And I thought, uh-oh. So I looked, and the browser had https://www.grc.sc/926. And there's no www. So that's an interesting sort of bad behavior on browsers because I never said www. But we do know that, and this is one of the things we've talked about, some web browsers have taken it upon themselves to probe www dot, you know, like to add that to the URL. Now, that won't resolve. On GRC's DNS there's no resolution to that. So you'd hope that the browser would back away.

Anyway, I'm not sure how the www dot got there. But for any of our listeners who may have had trouble, that was the shortcut for the registry, the little tiny registry reg file which turns off Windows querying and running any startup code when your machine boots, which are being supplied by the motherboard. So no www in front of the grc.sc/926. So thanks to Mrlinux for bringing that to my attention.

Andrew Drapper sent: "While the iPhone is quite locked down, the Mac less so. iMessage accounts are synced. Why can't these self-deleting messages be captured on a Mac?" And I thought that was a great observation. Now, the problem, of course, is that these attacks are highly targeted. So it'd have to be the coincidence of a user who was targeted having a Mac. And what we don't know is how long the actual attack took, that is, the iMessage arrives. It has an attachment. The attachment auto executes. It causes some other stuff to be downloaded from the command-and-control server. Those execute, go persistent, and then delete the attachment and the message. So that could be a few seconds; right? I mean, this is all blink blink blink blink, and now the malware is in place, and the message is gone. So presumably iMessage deletion also propagates, and there may not be any chance to grab it.

But if it is the case that deletion doesn't propagate, and somebody had a Mac synchronized to their phone, logged into their account, and was a target, then yes, the

planets could align, and maybe be possible to capture that. One has to wonder, though, you know, Leo, you sort of mentioned this last week, and I think you had a really good point. Apple might not add some quiet forensic stuff in order to like just capture stuff. The problem is they're end-to-end encrypted; right? They don't want to break their own users' privacy.

Leo: You know, I have to remember this. So this is the very important distinction. Messages are end-to-end encrypted unless you use iCloud storage for them. Then they are stored using a key that Apple has access to.

Steve: Right.

Leo: So that's the difference. When they say...

Steve: Unless we do that other thing that they just talked about; right? Where they did remove that last key that they have from iCloud so that they no longer have it. Historically they have. But there was something that we talked about, like a month ago.

Leo: Yeah. That was if you turned on Advanced Protection, though.

Steve: Right.

Leo: So...

Steve: I think that takes it out of their hands.

Leo: But it's not on by default.

Steve: Right. That is true. And it's not easy to turn it on.

Leo: It's not easy. And one of the reasons it's not is because you have to have everything up to date. Which tells me that there's some sort of key sharing, as you point out, key sharing mechanism built into the latest operating system.

Steve: That the older devices don't have.

Leo: That's right.

Steve: Yeah.

Leo: This is unclear. We talk about this sometimes on MacBreak Weekly because Apple's not fully forthcoming. But we know for instance, I mean, it's true of

Telegram, as well. Once you back up Telegram to iCloud, it's unencrypted, or it's available. We know because Apple provided Telegram messages, Paul Manafort's Telegram messages, to a subpoena to the law enforcement.

Steve: In response to a subpoena.

Leo: In response to a subpoena. So they have access to that. I think unless you turn on Apple's whatever they call it, Advanced Protection...

Steve: Yeah, the Advanced Data Protection.

Leo: Yeah. I bet you that those iCloud messages are encrypted but not end to end.

Steve: And then you've got to wonder how deep is the delete? Like when it deletes it, does the delete propagate?

Leo: Well, I would think so because they don't want to store stuff unnecessarily anyway, really.

Steve: That's my thought, too. That's my thought, too.

Leo: Yeah. But again, this is something - Apple kind of believes in security through obscurity. And so they're very hesitant to really reveal - maybe there's a whitepaper somewhere, but I've never seen it - exactly what's going on; you know? Yeah, it's an interesting question. I don't know. I'll ask around. Rene Ritchie used to be our expert on that, but not anymore.

Steve: Right, right. So, well, and they do maintain that beautiful security whitepaper. We used to go over it in detail.

Leo: Yeah.

Steve: But we've just been too busy lately.

Leo: Yeah. And that's a very good source. But it's my understanding that it's only end-to-end encrypted if you turn on Advanced Data Protection. I would guess, yes.

Steve: Yeah. So Jared Neaves says: "Hi, Steve. Hope you're well. I was wondering if you have any more book recommendations for us. I've enjoyed all the ones I've heard so far from you, but the last one I think I heard you talk about on Security Now! was the Bobiverse series, which was a while ago. P.S.: Just re-read one of my all-time favorites, 'The Mote in God's Eye' and its sequel, 'The Gripping Hand.' Amazing to think they conceptualized smartphones and AI home assistants back in the '70s."

Leo: Jerry was amazing. Yeah, he really knew his stuff.

Steve: Yup. That was the team of Jerry Pournelle and Larry Niven.

Leo: I often asked Jerry. He says, "I conceptualize." So the ideas come from Jerry, and then Larry would do most of the writing out, I think.

Steve: Yeah. Some of my favorite books of all time came from those guys.

Leo: So good, yes, so good, yeah.

Steve: So I'd already forgotten about the Bobiverse novels. They were definitely fun, and I read them all. We talked about them at the time. But somehow they didn't stick with me as much as some of the others, both old and new, have. So I'm just going to - I'm going to quickly bring people up to speed. The series that I most recently read was Scott Jucha's, and that's spelled J-U-C-H-A, his "Silver Ships" series. It was recommended to me by one of our listeners, and I am glad. There are 24 books in that series, 20 in the main line and four that are an offshoot, which then merge with the timeline as that original series of 20 finish. That whole series had some truly wonderful moments and many terrific new ideas. So I'm glad that I read all 24 of those.

There were an additional six books in his so-called "Gate Ghosts" series that caused me to do something that I almost never do, which was to quit without finishing. Now, this tendency of not quitting without finishing annoys my wife because we'll start in on some video streaming series which, after a few episodes, turns out not to be very good. She'll want to abort, but my inclination is to see it through to the end. I want to know. I'm into the story. I want to know what's going to happen.

Leo: I'm an optimist. I always think it's going to get better. It rarely does.

Steve: No. So it's very unusual for me to quit any sci-fi story in the middle, though I did quit the Gate Ghosts after I think two of its six books. I also, for that matter, quit Apple's "Foundation" series.

Leo: Yup, me, too.

Steve: To my wife's great relief. She says, "Oh, thank god we don't have to keep watching this."

Leo: It was pretty bad. Season 2 is imminent, by the way. You might want to finish it.

Steve: It was so disappointing. But I can't wait for the second half of...

Leo: Yes, finish Season 1 so that you can be disappointed by Season 2.

Steve: There we go, yes. "Dune," I can't wait for the second half of "Dune."

Leo: Oh, yeah. Lisa's saying, "Let's watch 'Dune 1' again because 'Dune 2' is coming up."

Steve: Yes, yes, yes, absolutely.

Leo: What a great movie that was, yes.

Steve: Anyway, the reason I dropped out of Gate Ghosts was a clear lack of action with no sign of any impending action. It turned into a mostly political narrative about the rights of sentient AIs and cloned humans as slaves. You know, I'm going to get bored after a while unless something blows up from time to time. Which leads me nicely back to my absolute favorite number one series in a long time, which is the Frontiers Saga by Ryk Brown, where Ryk is spelled R-Y-K.

Now, Ryk was a bit slowed down by a heart attack he suffered at 2:30 in the morning last December 4th. Fortunately, thanks to very good EMS response, he suffered almost no lasting cardiac damage. But the cause was severe blockages in three of his coronary arteries. So he underwent triple-bypass surgery on January 16th. Happily, his recovery was complete, and he picked up right where he left off with his ambitious plan for 75 full-length novels. And maybe we're going to get them all.

So while reading the Silver Ships series, I had fallen four books behind when I switched over to, well, as I said, the Silver Ships. Now I'm caught up. And if I may have been just a bit unsure after starting into this third 15-book arc, that was quickly dispelled once we got to book three. I would love to share a bit about what happens, but there are some fabulous surprises awaiting anyone who still has some catching up to do. I'll just say that in Ryk's work there is no lack of action, and plenty of things are blowing up all the time.

Leo: That's what I look for in a book.

Steve: Yeah. Got to have some stuff blow up. So those are the most recent two series. Earlier in this podcast I talked about the Honor Harrington novels by David Weber. There are 13 of those. They are wonderful, and she's a great character. There's also Jack Campbell's Lost Fleet series, which is also a ton of fun.

Nearly 10 years ago, in a PDF which is dated December 19th, 2013, so, yep, almost 10 years ago, I captured all of my favorite reading recommendations at the time, which we had discussed here on the podcast. I captured it into a Sci-Fi Novels Guide which just now, yesterday as I was writing this, I reviewed. It has the Honor Harrington series and the Lost Fleet series and all of our Peter F. Hamilton discoveries back then. So I've given it a shortcut using GRC's shortcut service, and it is - no www - it is just https://grc.sc/ - oh, and it just occurs to me, I could have done the sc from the dot sc. Anyway, it's .sc/scifi.

Leo: It would have been very confusing if you...

Steve: That would have been, yes, too confusing, grc dot scifi. So grc.sc/scifi, S-C-I-F-I. That will give you a PDF which is not just a list of books. It's also some commentary about how I felt about each book or series to sort of give you a little bit of guidance. And I'm going to eventually, at some point, need to update that because I would definitely like to add certainly the Frontiers Saga, which is just so much fun, and Silver Ships - although, Leo, I know you were not a fan of Silver Ships.

Leo: It was just a matter of style. I'm not a fan of - yeah.

Steve: Okay.

Leo: I don't have to have things blow up regularly in my books.

Steve: Oh, well, okay. As long as you don't mind when they do.

Leo: I don't mind when they do. Not too much, though. Just a little bit of blowing up.

Steve: You know, and all of those early Daniel Suarez novels, those were quite...

Leo: Whoa. "Daemon" had a lot of explosions. That was an action-rich novel, yeah, I agree. And I didn't mind that. It had a pace. It had a real pace to it, yeah. I just, you know, the thing I don't like about Silver Ships is just the hero. He's just too Captain America for me. I just can't...

Steve: I get it, yup. I do.

Leo: Too perfect. I like my heroes imperfect.

Steve: Well, speaking of which, I don't recall sharing any SpinRite testimonials since I began work on SpinRite 6.1. But we received one yesterday which I thought was interesting. Meir (M-E-I-R) in Montreal wrote, said: "Hello, Steve & Team. My son became a university lecturer and was very busy developing his course materials. After many hours of work, he discovered that the server he was using had not synced the data reliably, so he pulled out his backup 1TB USB memory stick, only to discover that the computer could not even see it. He asked me for help. I pulled out my copy of SpinRite 6.0; and it, too, did not identify the device.

"I then remembered that a few years ago you read a testimonial from a user who cooled his hard disk in a freezer. After three hours in the freezer, SpinRite had no problem seeing the USB device. It told me it would take 25 hours at Level 2, so I left it for the night. But in the morning, I saw this." And he sent me a picture of, and it's in the show notes, "Division Overflow Error. A critical error occurred at" - and I knew what it was going to say, B04E.

Leo: Uh-oh. How did you know that? How did you know that?

Steve: So he said: "Disappointed, I told myself maybe it recovered some of the data. And it did. All the 35 or so gigabytes of it. Freezing a USB stick? It worked. Thank you, Steve. Anxiously awaiting version 6.1. Meir in Montreal."

Okay, so first of all, the error that Meir encountered was that infamous problem in 6.0 that we now have a fix for. It's the result of SpinRite from 2004 encountering a drive 16 years later that it was never designed to handle. Thanks to the reverse engineering work of Paul Farrer, we have a simple patch utility that can be run before SpinRite 6.0 which will prevent this from ever occurring.

Secondarily, it is because of recoveries like this one that I decided to continue moving SpinRite forward even past 6.1. Although the days of spinning mass storage may be numbered, though that's been predicted for quite some time, and hard drives are still there and still make sense for many applications, I've previously shared many user stories of SpinRite recovering and repairing solid state drives. I remember initially being surprised by those reports. Now we just shrug and say, yeah, it works on those, too.

And what's even cooler is that, while not everyone always needs recovery, everyone could always use more speed. Over the weekend, one of our 6.1 testers from Australia, a guy named Peter Hancock, ran SpinRite on one of his thumb drives. I have a clip from the SpinRite log, this is SpinRite 6.1's log, which he posted in our GitLab. Among SpinRite 6.1's many new features is the ability to have it automatically run pre- and post-benchmarks on the drive. Although that option is not enabled by default, I wanted to provide this option since it helps SpinRite users to recognize that SpinRite is really not only about data recovery. This is clearly valuable maintenance for thumb drives.

So the numbers are in the show notes. Look at the read transfer rate at the front of Peter's drive. Before running SpinRite on it, it was reading at 1.077 megabytes per second, 1.077 megabytes per second. After running SpinRite it jumped to 14.021 megabytes per second, 13 times faster. And this is not some distant future SpinRite X. This is the free upgrade that everyone's getting. So dramatic improvement in the performance of solid state media. And this is not an isolated incident. Everybody who's running SpinRite on their SSDs is seeing this. And as for putting a thumb drive in the freezer, well, that's interesting, too. Meir in Montreal may have discovered something that will turn out to be useful for recovery. We'll have to see if we're able to repeat that experience, and then we'll add it to our tips and tricks for SpinRite.

Leo: Now we talk about scraping the Internet. There was just, it's interesting, I don't know if it's related, but there was just a story, came out today in Wired, saying that the United States intelligence divisions are illegally buying data broker information to spy on the entire populace.

Steve: Yeah, huge amount of...

Leo: Huge amount.

Steve: ...of proprietary independent data broker, yes, that was...

Leo: And it's technically not legal for them to do. The U.S. is openly stockpiling dirt on all citizens. The article today in Wired by Dell Cameron. But that's not exactly what you're talking about. You're talking about a different kind of scanning.

Steve: Correct. So with the news that Russia wants to block scanning of its internal networks and again, who could blame them for that? I thought it would be interesting to take a look at a modern state-of-the-art Internet scanning service, actually THE modern state-of-the-art Internet scanning service, to see where that state of the art is today. How quickly can the entire Internet be scanned? What ports are checked? What controls are available to be placed on such scanning? Can Roskomnadzor ask them to please not scan Russia?

So the story of the Internet's current state-of-the-art scanning begins 10 years ago, back in 2013, when a Turkish-American Ph.D. student at the University of Michigan named Zakir Durumeric looks at the existing, and at the time quite famous, though also quite an old kludge, NMAP network scanner and thinks, quite correctly, this could be done much better. Four years later, after being one of 14 students to receive a Google Ph.D. Fellowship in Security for the 2014-15 academic year, Dr. Durumeric finishes his Ph.D. thesis which is titled "Fast Internet-Wide Scanning: A New Security Perspective." The result of this work is a new scanner which Zakir names ZMap.

Okay, now, having created GRC's ShieldsUP! service back in 1999, which required the creation of an IP stack from scratch for that purpose, I know my way around packets, and I've seen a packet or two. So I can attest that Zakir's work is beautiful. His thesis demonstrates that he has an absolutely thorough grasp of many various problems, asks all the right questions, performs the right experiments, and winds up developing extremely robust whole-Internet scanning and assessment technology. It's really not rocket science, but no one had taken the time to sit down and really do it right until he did. And he did.

Now, in fairness to NMAP, which is many people have a strong fondness for, the world had changed dramatically since NMAP was first conceived. The biggest change was to the bandwidth available to such a scanner. At the time that Zakir came along, gigabit Internet connections were common and affordable. When I first created ShieldsUP!, my 1.54Mb T1 line was the envy of my friends. Since no scanner wants to create its own bandwidth denial of service on itself, the available bandwidth dictates everything else about the system's architecture. So Zakir was able to reconceptualize Internet-wide scanning at a time when doing so was feasible, like it really hadn't ever been before.

To gain an appreciation for the potential importance of the ability to have true near-real-time visibility into the Internet, I want to share the Introduction to Chapter 7 of Zakir's 216-page Ph.D. thesis. He wrote a book. It's a topic that all long-term Security Now! listeners will be able to relate to, since it happened on our watch. Zakir's Chapter 7 is titled "Understanding Heartbleed's Impact."

He writes: "In March 2014, researchers found a catastrophic vulnerability in OpenSSL, the cryptographic library used to secure connections in popular server products including Apache and Nginx. While OpenSSL has had several notable security issues during its 16-year history, this flaw, the Heartbleed vulnerability, was one of the most impactful. Heartbleed allows attackers to read sensitive memory from vulnerable servers, potentially including cryptographic keys, login credentials, and other private data. Exacerbating its severity, the bug is simple to understand and exploit.

"In this work, we analyze the impact of the vulnerability and track the server operator community's responses. Using extensive active scanning, we assess who was vulnerable, characterizing Heartbleed's scope across popular HTTPS websites and the full IPv4 address space. We also survey the range of protocols and server products affected. We estimate that 24 to 55% of HTTPS servers in the Alexa Top 1 Million were initially vulnerable, including 44 of the Alexa Top 100.

"Two days after disclosure, we observed that 11% of the HTTPS sites in the Alexa Top 1 Million remained vulnerable, as did 6% of all HTTPS servers in the public IPv4 address space. We find that vulnerable hosts were not randomly distributed, with more than 50% located in only 10 ASes - those are top-level ISPs, right, AS is Autonomous System number - 50% located in only 10 ASes that do not reflect the ASes with the most HTTPS hosts.

"In our scans of the IPv4 address space, we identify over 70 models of vulnerable embedded devices and software packages. We also observe that both SMTP plus TLS and Tor were heavily affected. More than half of all Tor nodes were vulnerable in the days following disclosure.

"Our investigation of the operator community's response finds that, within the first 24 hours, all but five of the Top Alexa 100 sites were patched; and within 48 hours, all of the vulnerable hosts in the Top 500 were patched. While popular sites responded quickly, we observed that patching plateaued after about two weeks, and 3% of HTTPS sites in the Alexa Top 1 Million remained vulnerable almost two months after disclosure."

Okay. Now, think about how valuable it is to have this sort of information in the wake of a significant Internet-wide security event like Heartbleed. When this happened in 2014, there was no other source of this information, other than Zakir's new Internet scanner, which he had written as a freshman at University of Michigan. There's only one way to get this kind of information, which is to have the tools that are able to go out onto the Internet and look at it. And they need to be fast.

Okay. So Chapter 4 of Zakir's thesis, which I won't go into here, is titled "Detecting Widespread Weak Keys in Network Devices." It's another example of how crucial having this sort of visibility into the Internet can be. A new vulnerability and/or attack is discovered on some core aspect of our global Internet, and we need to be able to assess its impact and to begin to know how to remediate its effects. Where to look. Who to call. What to do.

In an interview which Zakir gave to the Turkish American Scientists & Scholars Association, he was asked: "Could you describe your innovation in layman's terms, and how it relates to everyday life?" He replied: "The cornerstone of this research is ZMap, a tool that I introduced in 2013 that enables researchers to rapidly measure how every device connected to the public Internet is configured. ZMap reduces the time required to perform Internet-wide measurements from months to minutes, 10,000 times faster than previous techniques, and allows us to reason about the devices that make up the Internet for the first time. Previously, many decisions were made anecdotally or through sampling. Now we're able to perform comprehensive measurements, which has allowed us to uncover new types of bugs and understand some of the more complex interactions between devices at scale."

Okay, now, what happened to Zakir? Today he's an Assistant Professor of Computer Science at Stanford University, and Chief Scientist of Censys, which is the inevitable commercial spin-off of his work. But before we get to that, let's look at the non-commercial side, which is the ZMap Project. It's located at zmap.io. And no www in front of that, either. Just zmap.io.

The ZMap project describes itself as "a collection of open source tools" - all of this is open source - "open source tools for performing large-scale studies of hosts and services on the Internet. The project was started in 2013 with the release of ZMap, a fast single-packet scanner that enabled scanning the entire public IPv4 address space on a single port in under 45 minutes. A year later, we released ZGrab, a Go application-layer scanner that works in tandem with ZMap. Since then, the team has expanded, and we have built nearly a dozen open source tools and libraries for performing large-scale

Internet measurements. Continued development is supported by the National Science Foundation, the U.S. NSF."

So the Project has published a series of papers that describe how the suite of ZMap tools are designed. There's ZMap: Fast Internet-Wide Scanning and its Security Applications. There's another paper, ZDNS: A Fast DNS Toolkit for Internet Measurement. There's ZLint: Tracking Certificate Misissuance in the Wild. And LZR: Identifying Unexpected Internet Services. And I found this one particularly interesting. That last paper, the LZR, identifying unexpected Internet services, was delivered during the 2021 USENIX Security Symposium. The synopsis of this "Identifying Unexpected Internet Services" paper has a couple of surprising findings.

It says: "Internet-wide scanning is a commonly used research technique that has helped uncover real-world attacks." Well, it's common now. It wasn't common in 2013. "Helps uncover real-world attacks, find cryptographic weaknesses, and understand both operator and miscreant behavior. Studies that employ scanning have largely assumed that services are hosted on their IANA-assigned ports." Right? Like HTTP, port 80; HTTPS, port 443; Telnet, port 23; and so on. Those are the IANA-assigned ports. The idea being that if you're given the IP of a service, you know what the service is, you know which port it's expected to be on at that IP address.

So they go on: "Overlooking the study of services on unused ports. In this work, we investigate where Internet services are deployed in practice and evaluate the security posture of services on unexpected ports. We show protocol development is more diffuse than previously known, and that protocols run on many additional ports beyond their primary IANA assignment. For example," they say - get this - "only 3% of HTTP and 6% of HTTPS services run on ports 80 and 443." What? 3% of HTTP is on port 80, and only 6% of HTTPS is on 443.

Leo: Well, where else is it?

Steve: Everywhere else. Like on the other 65534 ports.

Leo: But it starts on 80.

Steve: No. Well, like we know, Leo, that to run a server on port 80 you have to have root. Right?

Leo: Right.

Steve: Because you can only run - you're only able to access ports below 1024, that is, 1 to 1023, if you have root privilege. That's why userland servers often use port 8080.

Leo: 8080, right, right.

Steve: As their HTTP. So the point is these services are scattered all over hell and gone.

Leo: The public-facing servers have to use 80 and 443. Otherwise you wouldn't get there.

Steve: Well...

Leo: No?

Steve: Or you have to know to put a colon something after the URL.

Leo: Right.

Steve: Which, you know, sometimes happens.

Leo: Sometimes, sure.

Steve: You will sometimes see a URL that says, you know, :12345. It's like, okay. So they conclude: "Services on non-standard ports are more likely to be insecure, which results in studies dramatically underestimating the security posture of Internet hosts. Building on our observations, we introduced LZR ('Laser'), a system that identifies 99% of identifiable unexpected services using five handshakes and dramatically reduces the time needed to perform application-layer scans on ports with few responsive expected services." And they give an example of having achieved a 5500% speedup identifying MongoDB on port 27017.

Okay. So again, who would have imagined that only 3% of all of the actual HTTP service is on port 80? The other 97% are on other ports. And similarly...

Leo: I think it's misleading. All public HTTP servers are on 80, and HTTPS servers are on 443. Right?

Steve: No. This is a scan of the public Internet. All of this is the public Internet.

Leo: Yeah, but like when I set up a localhost thing, it's on a different port, you know, when I'm writing stuff for a web server or whatever.

Steve: Right.

Leo: But my website has to be on port 80; doesn't it?

Steve: Yes. But other people can have other things on other ports.

Leo: But I'm not surprised they're insecure because those are often just like somebody messing around and stuff.

Steve: Or, you know, what's that horrible NAS that keeps getting hacked?

Leo: QNAP?

Steve: Yeah, QNAP. It's probably got a web server running on some port; right? So there's a public service running on a non-80 port, for example.

Leo: It shouldn't shock you, though, if they're running on non-canonical ports, that they might be insecure, too, because, like, why; right?

Steve: And the point is we never looked. We never knew.

Leo: Right.

Steve: So what happened was, when we looked, it was like, oh, my god, look at all this crap everywhere else that is, like, listening on these bizarre ports that no one ever thought to look at.

Leo: Right.

Steve: So two papers later, the ZMap Project's list of published papers, we encounter Censys, C-E-N-S-Y-S. "Censys: A Search Engine Backed by Internet-Wide Scanning." And the synopsis of this paper shows us how we move from Internet-wide scanner to an Internet-wide search engine. And just so that I'm sure I have everyone's attention, this is free. So search.censys.io, you can put some search terms in and pull from what I'm about to be reading. And you don't have to be a university researcher. You just have to be researching for a good purpose, and they will give you credentials. There is a commercial side where like all the big security companies, they pay to have API backend access to this database. But people listening to this podcast could say, hey, I want to - I've got an idea. I want to do some research, and you can get signed up.

So here's the deal. They said: "Fast Internet-wide scanning has opened new avenues for security research, ranging from uncovered widespread vulnerabilities in random number generators to tracking the evolving impact of Heartbleed. However, this technique still requires significant effort. Even simple questions, such as 'What models of embedded devices prefer CBC ciphers?' require developing an application scanner, manually identifying and tagging devices, negotiating with network administrators, and responding to abuse complaints.

"In this paper, we introduce Censys, a public search engine and data processing facility backed by data collected from ongoing Internet-wide scans. Designed to help researchers answer security-related questions, Censys supports full-text searches on protocol banners and querying a wide range of derived fields. It can identify specific vulnerable devices and networks and generate statistical reports on broad usage patterns and trends. Censys returns these results in sub-second time, dramatically reducing the effort of understanding the hosts that comprise the Internet. We present the search engine architecture and experimentally evaluate its performance in this paper. We also explore Censys's applications and show how recent questions become simple to answer."

And this brings us to the second part of this, which is censys.io. The Censys mission statement reads: "At Censys we believe that cybersecurity is critical to the future of our global economy. And in order to evolve cybersecurity defenses, both the public and private sector need access to best-in-class intelligence data. By arming our customers with the visibility and insights that they need to protect against critical threats, Censys provides the intelligence needed to bolster cybersecurity capabilities worldwide." So basically we now with Censys have access to near-real-time, deep intelligence, Internet-wide, port-wide scanning of everything that is public.

What does Censys tell us about their Internet scanning? Their timeline notes that ZMap was invented in 2013, which we know. Zakir did that. Thus Z of ZMap. That Censys was founded four years later in Ann Arbor, Michigan, probably when he got his bachelor's, which is where Zakir had gone to University. And it also shows that two years later, in 2019, the original ZMap scanner was replaced by their proprietary scanning technology. So there are two projects. There's the ZMap project which has continued moving forward, advancing ZMap scanner-based applications, which is still open source and available. And then there's Censys, which has moved to a far more sophisticated scanning technology. We don't need that because we have access to its results. And of course it makes sense that there would be a change over six years. As I said, the world is constantly changing. And six years since Zakir first created ZMap, you can now do a much better job. And so that's what they have.

Under the topic of "Host Scanning Introduction" they explain: "Censys continually scans the entire public IPv4 address space on 3,592-plus ports using automatic protocol detection to present the most accurate representation of the Internet's current state. Censys also leverages redirects and the Domain Name System to discover and scan around 79 million in-use IPv6 addresses." Now, that's interesting, since it's entirely possible to scan all IPv4 addresses which occupy a 32-bit address space. However, there's no possibility of scanning the entire IPv6 address space, which is 128 bits. So it's necessary instead to discover, hold, and build up a sparsely-populated map over time of active IPv6 addresses, which these guys have. And it's also interesting that this number is still apparently as few as around 79 million. Okay. 79 million is a lot, but it's not close to IPv4's essentially fully occupied 4.3 billion IPs.

They continue to explain: "Censys scans only obtain information. Censys never attempts to log into any service, read any database, or otherwise gain authenticated access to any system." They ask themselves the question: "How often does Censys scan for new services? Discovery means finding a service on an IP and port that was not there last time we looked. Censys has several schedules for discovery based on our experience scanning the Internet. First, Global Scan of Popular Ports: We scan the whole IPv4 space on 137 ports with IANA-assigned services every day." Okay? So the entire IPv4 space on all 137 IANA-assigned ports every day.

"Cloud Provider Scans: Since many cloud hosts are ephemeral, we scan the 1,440 most popular ports on Amazon, Google, and Azure hosts every day. Global Scan of Less Popular Ports: We scan the whole IPv4 space on 3,455 additional ports on a regular basis, completing a walk every 10 days. And finally, a Global Scan on Every Other Port Number. We scan the entire IPv4 address space across ALL ports (65535) at a low background rate."

Okay. So stop for a second. Internet background radiation? Yeah. That means any IP address, any IPv4 address. You put a port monitor on it, and these guys are going to wind up sending a packet to you to every single one of your IP, your one IPs, 65535 ports, over some period of time. All 137 of the IANA-assigned ports every day, on every single IP out there. So this is some radiation.

"How Often Does Censys Refresh Data for Known Services? Once a service has been discovered, Censys prioritizes refreshing the information about that service to ensure it is accurate and up to date. Once a day, the age of each of the approximately 2.1 billion services in our data set is checked. Any unnamed service with an observation timestamp older than 24 hours is rescanned. With this process, the average age of high-value service data is about 16 hours." Okay. So that means they've got 2.1 billion specific services, meaning a port, an IP, and they know what's there, 2.1 billion of them, and they revisit it and reverify it about every 16 hours. None of the data in that data set is older than 16 hours. And it's available to be searched.

"How Does Censys Scan?" Okay, now, they don't really answer their own question, which I thought was interesting, even though they asked it of themselves. They say: "Censys has invested time and technology into setting up multiple global perspectives and developing sophisticated scanning techniques to produce the richest, most useful data set for the security community." Okay, well, that didn't really answer the question. They could have just said "We scan good."

Anyway, they said: "Censys peers with and scans from five Tier-1 ISPs (NTT, Tata, Hurricane Electric, Telia, and Orange) to produce nearly 99% coverage of listening hosts across the globe with enhanced protection against packet drop. The ISP that Censys scanned any given service from is recorded in the services.perspective field.

"Deep Protocol Scans: On ports with IANA-assigned protocols, Censys attempts to complete a handshake with the assigned protocol," you know, expecting Telnet to be on port 23, for example. They said: "If that fails, we try additional handshakes according to our experience with protocol and port pairings." Okay, so, for example, if a Telnet handshake were to fail on port 23, they might try an SSH handshake since its foreseeable that someone might have put SSH over on Telnet's port. They said: "On ports without an assigned service, we start by sending an HTTP request" - because, as we've seen, HTTP is kind of everywhere - "and attempt to automatically detect the protocol based on the response.

"Automatic Protocol Detection," they said. "The Censys scanner analyzes every server response to identify its service, even if it's non-standard for the port, which allows us to uncover the vast majority of services in unexpected places. For example, if an HTTP request results in an SSH banner, Censys will close the HTTP connection and reattempt an SSH handshake. Censys can detect 25 protocols on any port. Some protocols do not have a lot of data to parse and index." Meaning, if an initial TCP connection handshake succeeds on some random unassigned port, now what? So they said: "Censys identifies 47 lightweight services and collects a banner.

"What Protocols and Services Does Censys Detect? Censys can detect and complete scans for over 100 Layer 7" - meaning application layer - "protocols. The default Layer 4 [IP layer] protocol used by our scanners is TCP, although some protocols, such as DNS, are scanned with UDP, and HTTP can be detected over QUIC. Service names represent the most specific service information we have. For example, a generic HTTP service has a service name of HTTP, while an HTTP service that's actually an Elasticsearch server has a service name of ELASTICSEARCH." Meaning that they will get as specific as they can.

"There's also an UNKNOWN fallback, which means that Censys could not identify the protocol in use by an open service, either because the service is not adhering to a protocol" - and they say there are a lot of HTTP-like services out there - "or because Censys does not have a protocol-specific scanner written for that. So once a day around 2.1 billion individual Internet services which they maintain in their data set is checked, and they maintain a searchable index of everything that they have found."

And what if someone like Russia doesn't want to be scanned and indexed? About this Censys says: "Censys strives to be a good citizen of the security industry. We never attempt to log in to any service, never read any database, or otherwise gain authenticated access to any system.

"Can I opt out of Censys data collection?" They said: "Censys scans help" - like they don't want you to. So they say: "Censys scans help the scientific community accurately study the Internet. The data Censys gathers is sometimes used to detect security problems and to inform operators of vulnerable systems so that they can be fixed. If you opt out of data collection, you might not receive these important security notifications. However, if you wish to opt out, you can configure your firewall to drop traffic from the subnets we use for scanning."

And here in the show notes and on their site they proudly display them. There are five /24 networks, so that's 256 IPs, or 253 or 4. So there are five of those networks. And then two IPv6 networks which are /80s. So, what, 48 bits on the machine name. And so again, you're not going to scan all those. But so they do show a total of seven networks which, if you didn't want them, if you didn't want your network to be in their database, in their data set, which could be searched for things, add those to your firewall.

Okay, then what happens? They did say: "Additionally our HTTP-based scans," because they do make HTTP queries to see what's there, like on port 80, "use a Censys-specific user agent, which can be used to filter requests from our scanners." So again, if you arrange to have your server reply with, it's got CensysInspect/1.1 - I'm sorry. If you see this coming as the user-agent into your server, you could have your server do something different, just drop the request, you know, just hang up.

They said: "Configuring your services to drop connections from Censys's subnets will prevent our scanners from indexing your services. Historical data is not removed from Censys data sets as part of this change. Host services are typically pruned from Censys Search within 24-48 hours of their last observation timestamp." So they're not just accruing everything forever. If you go off the 'Net, after a couple days of you not being there, they will remove you because they want to be a current snapshot, not an archival snapshot. They said: "Host services are typically pruned from Censys Search within 24-48 hours of their last observation timestamp, while Virtual Host services can remain in the data set for up to 30 days."

And finally they explain: "Censys started as a research project at the University of Michigan, and we continue to provide free Internet data to the research community. We provide verified researchers" - and the verify, as I said, the verify threshold, the verify bar is very low. Our listeners could qualify - "the same access to our data as our highest-tiered commercial customers." I've got a link to the research access to Censys data, here on the last page of the show notes.

So as I said, while they do sell access to their databases and data sets to commercial entities, which they make available through an API, they also make this access available to pretty much anyone who has a justifiable use for such access. And they say, you know, if your use is I want to make my company's hosts more secure, no. But if you want to make the world a better place, welcome, and we'd be happy to have you.

So I've been noticing that we've been running across this group, the name Censys, more and more, with their name being cited by other security researchers who are clearly their commercial customers. So I've been wanting to do a bit of a deep dive into who they are and where they came from. Now we know. So the next time I refer to them, it won't be, "Who?" It'll be, "Oh yeah, those guys. They're good." I believe they are.

Leo: This is fascinating. And well done to Censys, I guess; yes?

Steve: Yeah. I mean, to be giving the world a near, I mean, as real-time as it could be, maintained, searchable data set of what is the Internet. I mean, that is. That is the Internet.

Leo: Could it be misused in the way that Shodan is kind of misused?

Steve: Absolutely.

Leo: Okay. Just checking.

Steve: Absolutely.

Leo: It's like Shodan, basically.

Steve: Yeah.

Leo: Yeah. Okay.

Steve: Yeah. I mean, you know, they used that access to find and enumerate the Heartbleed-vulnerable servers immediately.

Leo: Right, right. So there you go.

Steve: So, yeah.

Leo: Wow, fascinating. And another great show, jam-packed episode. This is a two-hour, 21-minute extravaganza. Well done, Mr. Gibson. Steve's at GRC.com. That's where SpinRite lives, the world's best mass storage maintenance and recovery utility. 6.0 is going to be 6.1 any day now. Buy 6.0 today, you get to use it. And as soon as 6.1 comes out, you'll get that for free, a free upgrade. GRC.com. While you're there, pick up SpinRite.