Operation Triangle

Description: Today's podcast is chock full of news. What has DuckDuckGo just announced? What about the Tor Project? Has Opera just made a big mistake? What is the KasperskyOS? What's happening to non-Russian web hosting for Russians? Are SolarWinds executives finally going to be held to account? We now have the U.S. Space Force, what's next? What's the latest large site to support Passkeys? Who would like permission to spy on their own citizens? Which facial recognition smartphone unlocking can you trust and which should not be? And what was the inevitable shoe to drop following last week's coverage of the Massive MOVEit Transfer mess? Then, after sharing a bit of listener feedback, we're going to take a much closer look into Kaspersky's discovery of a pervasive 4-year iPhone spyware campaign.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-929.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-929-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got new browsers from DuckDuckGo and Opera, now with AI. We'll also talk about Kaspersky's discovery of a severe bug on iPhones - that's why there was an Apple update, emergency update last week - and the cost of doing business in the Russian federation. That and a whole lot more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 929, recorded Tuesday, June 27th, 2023: Operation Triangulation.

It's time for Security Now!, the show where we cover the latest security news, of which there was a lot, and I missed. But fortunately, Steve's going to fill me in. Hi, Steve Gibson.

Steve Gibson: Yes, Leo, welcome back. We missed you last week. We had a - it's funny, too, because throughout the podcast I recognized how much continuity there is in this podcast, and so I was...

Leo: We build on, each episode builds on the episode before.

Steve: Yes. And so I kept saying to Jason, well, you wouldn't know this, but blah blah blah blah, and like trying to quickly bring him into what we were talking about. But then I thought, well, wait, doesn't he produce this podcast? So maybe he's being forced to listen to it every week whether he wants to or not.

Leo: I think he may be forced to listen to it.

Steve: Yeah, yeah.

Leo: Anyway, thank you, Jason Howell, for filling in. I really appreciate it. And we had a good time in L.A.

Steve: So then I thought, maybe this is dumb that I keep saying, well, you wouldn't know this, if he's thinking, yes, Gibson, unfortunately I do.

Leo: I know it. You know, it's a fair thing to say about anything you say on this show, "You wouldn't know this," because it's advanced. These are advanced topics.

Steve: But I actually created the second page of our show notes for you, Leo, because I just - I couldn't have you miss the things that we talked about, mostly because there were only three.

Leo: Yes, I did note that, yes.

Steve: Okay. But so this week we're going to come back to a topic that we opened three weeks ago, talking about Operation Triangulation. And at the very end Eugene Kaspersky tells us why he named it that, which I hadn't seen anywhere else.

So today's podcast is chock full of news. We're going to answer a bunch of questions. What has DuckDuckGo just announced? What about the Tor Project? Has Opera just made a big mistake? What is the KasperskyOS? What's happening to non-Russian web hosting for Russians? Are SolarWinds executives finally going to be held to account? We now have the U.S. Space Force. What's coming next? What's the latest large site to support Passkeys? Who would like permission to spy on their own citizens? And that's a little disturbing.

Which facial recognition smartphone unlocking can you trust, and which should not be trusted? And what was the inevitable shoe to drop following last week's coverage of this Massive MOVEit Transfer mess? Then, after sharing a bit of listener feedback, we're going to take a much closer look at Kaspersky's discovery of a pervasive four-year-long iPhone spyware campaign. And Leo, we've got another great Picture of the Week and a recap of last week's because I couldn't have you miss it.

Leo: Steve, I am touched and honored that you care so much about me that you did that. That is very sweet. I appreciate that. I have not looked. I have not peeked at any of our Pictures of the Week, either of our Pictures of the Week.

Steve: Good, because this one, the concept behind the picture was not unique, but I'm happy with the caption that I gave it. And it ends up creating a wonderful effect for people who are in the know, as of course you and our listeners all are.

Leo: Yes. You know, you raise a really interesting point. I hadn't really thought about it, but it is true. You probably don't want to miss an episode of this show, and I hope none of our listeners are. Subscribe and download, and that way you'll always have a copy because we do build, each show does build, assume a certain amount of knowledge.

Steve: Yeah.

Leo: You've got mail, Steve. I hear it. I hear it.

Steve: I forgot to silence my phone next to me.

Leo: Bamm-Bamm. So is it Picture of the Week time?

Steve: Picture of the Week time. And it's fun.

Leo: All right. Should I show the picture?

Steve: Sure.

Leo: I'll pull it up. I'm going to see it first. And then I will switch over to it so that you can - okay. Very good. Very funny. The caption you wrote; right?

Steve: I wrote the caption, yeah.

Leo: All right. Let me show it here. Unfortunately it keeps shrinking down. I want to show it full screen, if I can.

Steve: But I think it was Simon Zerafa, our longtime podcast listener and follower, who had the concept of it. And I thought it was very clever.

Leo: When Apple creates a walled garden, they don't mess around. And of course what's the picture of?

Steve: Yeah, Apple headquarters, which is enclosing a beautiful garden, and that's one hell of a wall.

Leo: It is a literal walled garden, yeah.

Steve: It actually is, yes.

Leo: Yeah.

Steve: I thought that was very cool.

Leo: A lot of people pointed out that the Apple campus, the brand new headquarters of Apple, a few years old, faces inward, not outward. That's another point to mention.

Steve: Yeah, yeah. And like we don't, like normal people don't ever get to see it, do they.

Leo: No. No.

Steve: I mean, do they give - it's just completely off limits to everyone.

Leo: We get to go to the Visitors Center out here. Sometimes people get to go inside the ring. But, yeah, you really have to be somebody special to be invited inside, inside the special spaceship.

Steve: We see little bits of it on the WWDC, and it's just spectacular.

Leo: Yeah. It really does look beautiful. I mean, when you've got the money. And Jony Ive designed it with the help of Steve Jobs. You kind of get the best.

Steve: Yeah. Okay. So now the next page of the show notes is the security catch-up for you, Leo.

Leo: Okay. Starting with the Picture of the Week.

Steve: Starting with last week's Picture of the Week.

Leo: If it's not tied down, of course, all that's left of this bicycle - which has been triple-locked, two U-locks on the wheels, and even a lock around the seat. And of course all that's left is the wheels and the seat. The frame is long gone. Yeah, that's the problem. Bicycle thieves, I tell you.

Steve: I just love it. So, okay. So the few subjects that we talked about last week, believe this or not, and you just had to know this, Leo, which is why I didn't want you to miss it.

Leo: Okay.

Steve: It turns out that those brilliant researchers at Israel's Ben-Gurion University of the Negev, who are always coming up with wacky ways of exfiltrating data, they listen to Security Now!.

Leo: Whoa.

Steve: And the lead researcher sent me a DM saying, hey, Steve, we've just finished some other research. You know, love the podcast.

Leo: Wow. We talk about them all the time.

Steve: We do. And they're listeners. So get this. You know how much fun we've had joking about the fact that the flashing LEDs on our routers don't actually convey any data. All they're doing is showing that there's something going on on the wire. Well, that's true. It turns out, however, that they have discovered that the power LEDs of equipment doing, like, secret computation is affected enough by the work required to process the secret information in crypto algorithms that they have been able to recover secret keys by recording the LED fluctuations. And they're even able to do it using an Apple iPhone 13.

Leo: Oh, man.

Steve: Because even though the frame rate of the iPhone is 60Hz, it turns out that the imaging arrays are scanning at 60Hz. So if you fill the frame with the image of an LED, like by zooming in or holding the camera really close to it, you end up getting two orders of magnitude. So it goes from 60Hz to 60KHz of effective scanning rate. And that gives them enough of a high sample rate to be able to capture minor intensity fluctuations which they've been able to then reverse engineer the secrets that are being processed by the equipment.

Leo: Wow. Wow.

Steve: So you just had to know that. While you were off riding the Dumbo ride...

Leo: That's amazing, yeah.

Steve: ...we were here doing real work.

Leo: I did ride the Dumbo ride. You must have seen our Instagram post.

Steve: I'm sorry. Well, I heard you did, please, what about it's a Small Small World?

Leo: I did not go on it. I did not. But I did go on the Dumbo ride.

Steve: Okay. That's okay.

Leo: I'll show you. That'll be my Picture of the Week a little later.

Steve: The last thing is that, believe it or not, we were hit, actually the entire industry was hit with the 25-year-old persistent SQL injection vulnerability.

Leo: Oh, my gosh.

Steve: I know. I was going to say hundreds, but it's actually a couple thousand companies were all using some software from a company called Progress Software called MOVEit Transfer.

Leo: Oh, yeah, yeah, yeah.

Steve: Which is a managed file sharing facility. Turns out horrible SQL injection vulnerabilities. All of them got their data exfiltrated and are now being extorted. And I mean the list of companies that this happened to was astonishing. And so of course that sent me off on a tirade last week about how is it that in 1998 this was observed as being a problem, Apple said don't worry about it, it's not a problem, and here we are now. You know, it's also been OWASP's number one security threat, like, constantly on their list. And it happened again.

Oh, and the final announcement of last week was that SpinRite is at, like, completion. I proposed, I released Alpha 29, and then Alpha 30. We're, like, right at the edge of this thing being done. It looks like I may actually have just broken something in the last couple days. But, you know, a so-called regression.

Leo: Hands off, Steve. Hands off.

Steve: I'll get that fixed.

Leo: Stop touching it. And here I am, while you were talking about that, literally on Dumbo.

Steve: And there's Lisa in her own - is Lisa behind you there?

Leo: Yeah, she had her own Dumbo. We didn't want to share.

Steve: She had her own Dumbo car, okay.

Leo: Yeah, yeah, well, yeah.

Steve: It looks like to me you're having too much fun on the Dumbo ride, Leo.

Leo: It was a lot of fun, I have to admit.

Steve: What?

Leo: That's more my speed than the roller coasters. She went on the Matterhorn. I did not.

Steve: Wow. Of course, none of the roller coasters at Disneyland are worth anything.

Leo: Yeah, they're not that scary.

Steve: You need Knott's Berry Farm kind of stuff.

Leo: Oh, you're a purist, are you? Okay.

Steve: Oh, yeah.

Leo: You know that the - they used to call it California Screamin'. It's the Incredicoaster at the California Adventure part of Disneyland in L.A. Just down the road from you, I might add. Has a rollercoaster that is a railgun. Did you know that?

Steve: I loved seeing the pictures of it. You were talking about it somewhere. And I think that's so cool.

Leo: It's magnetic conduction. They have these plates, acceleration. They have these plates in the car so you get positioned there, instead of, you know, most roller coasters clickety-clack up a steep hill, and you know there's a lot of anticipation. When we get to the top of the hill, we're going down. This one's...

Steve: I grew up at Santa Cruz, so I know all about...

Leo: Oh, the worst boardwalk - that thing, that was the thing, that's why I don't go on roller coasters. I was in high school.

Steve: It's rickety wood. And boy, you just...

Leo: And the beams are going by you 80 miles an hour, inches from your head. And that's the first and last roller coaster I ever went on, in high school.

Steve: The world has changed a lot. You used to have Easy-Bake ovens where you could burn your fingers.

Leo: Yes.

Steve: And you used to have chemistry sets where you could actually create chlorine gas.

Leo: Yes.

Steve: I actually did, so I know that.

Leo: Oh, geez.

Steve: But not anymore.

Leo: Oops.

Steve: Now, no, children, you cannot have anything that's actually fun to play with.

Leo: No.

Steve: No actual rockets that work.

Leo: Yeah, the roller coasters are safer now. That's the upside, yeah.

Steve: So DuckDuckBrowse. Joining the macOS browser they launched last year, DuckDuckGo now has their Windows browser in public beta.

Leo: Ah, good.

Steve: As we would expect from the privacy-first search folks, the DuckDuckBrowser and I sure do hope that's not what they're going to actually name the thing is privacy first. It sports, and I kid you not, the Duck Player, which is a YouTube player that allows viewing YouTube videos without privacy-invading tracking ads.

Leo: What?

Steve: Yes, and prevents videos viewed from impacting future recommendations. So they're not letting it track you and profile you.

Leo: Enjoy it while you can, kids, because that's not going to last.

Steve: No.

Leo: YouTube has already blocked most adblockers on YouTube. I think it's just a matter of time.

Steve: So the Duck Browser may not be long for this world. They claim that the browser's tracker blocking, which is built in, goes above and beyond what's available from Chrome and other browsers. They wrote: "Our third-party Tracker Loading Protection, for example, blocks the hidden trackers from companies like Google and Facebook lurking on other websites before they get a chance to load." And it's unclear what this means. They wrote: "Smarter Encryption to ensure that more of the websites you visit and the links you click are encrypted, relative to other browsers."

Leo: Okay. So this was written not by the engineers, but by the marketing department, I'm certain. That makes no sense.

Steve: Yeah, yeah. I guess what they're saying is they're being more clever about choosing HTTPS alternatives when those are available. But really that problem's kind of been solved already, so I'm not sure that that's that useful. Now, okay. Here's something that might be worth the price of admission, which being zero admittedly sets the bar rather low. But they said: "Cookie Pop-up Management, a tool that automatically selects the most private options available and hides cookie consent pop-ups."

Leo: Yes. Yes.

Steve: So I would like to have that. That would be good. I don't know how they'd do it because how are you going to, like, automatically respond to arbitrary pop-ups and choose the most private one? I don't know.

Okay. Now, here's a problem. They called it the Fire Button, as in lighting a fire. And they said: "Burns recent browsing data in one click."

Leo: Oh, please.

Steve: I know. And there's also the Fireproof option, that's what it's called, for any sites you want to stay logged into. Now, I suppose if you name your privacy-centric search service DuckDuckGo, then you've already lowered expectations.

Leo: We know you hate that name.

Steve: Oh, my god, about the name you're going to use for other things. But somehow the idea of a web browser having a "burn bag" into which websites are tossed by pressing the "Fire" button, to light them on fire and reduce them to ashes unless you have "Fireproofed" them ahead of time - I don't know. Maybe it wasn't the marketing

people after all, Leo, because this really seems like it should not have gotten out in the public view. The browser also offers built-in email protection to hide user email addresses behind uniquely generated @duck.com, because that's what everyone wants to be known as, addresses when signing up online. Now, while that sounds handy, it also would create some quite powerful lock-in effects, if like all of your logins are some email address @duck.com. So I'm not sure about that.

Anyway, the beta of the browser, which apparently goes by the catchy name "DuckDuckGo for Windows," is available from, not surprisingly, duckduckgo.com/windows. And they note that switching is easy since of course like all current browsers it's able to import bookmarks and passwords from other browsers and password managers. Their announcement had a couple of additional interesting things to say.

They wrote: "The browser doesn't have extension support yet, but we plan to add it in the future." And I would say, well, okay, if it survives. Anyway, they said: "In the meantime, we've built the browser to include features that meet the same needs as the most popular extensions, ad blocking and secure password management."

So they said, of secure password management: "Our browser includes our own secure and easy-to-use password manager that can automatically remember and fill in login credentials. DuckDuckGo for Windows can now also suggest secure passwords for new logins." Which of course everybody else has already had for a decade. "This will get even more convenient soon when we roll out private syncing across devices," which, you know, you really can't use this until it has that. "So you'll be able to sync your bookmarks and saved passwords between different devices, whether you're using a DuckDuckGo browser on Windows, iOS, Android, or Mac."

Okay. Ad blocking: "DuckDuckGo for Windows is equipped with our privacy-protecting alternative to ad blockers. The browser blocks invasive trackers before they load, effectively eliminating ads that rely on creepy tracking." You know, because, they said, "So many ads work that way, you'll see way fewer ads, if any at all. We also remove the whitespace left behind by those ads..."

Leo: Well, that's good.

Steve: Yeah, "...for a clean, distraction-free look without the need for an outside ad blocker." So, yeah, that sounds good. And finally: "Duck Player, our browser's more private way to watch YouTube." They said: "This built-in video player protects you from tracking cookies and personalized ads with a distraction-free interface that incorporates YouTube's strictest privacy settings for embedded video."

They said: "In our testing, by blocking the trackers behind personalized ads, Duck Player prevented ads from loading on most videos altogether." Which again, Leo, I agree with you, like let's see how long this lasts. "YouTube still logs video views, so it's not completely anonymous. But none of the videos you watch in Duck Player contribute to your personalized recommendations or your YouTube advertising profile. You can leave the feature always on, or opt in on individual videos."

And I thought that what was most interesting was that this recently created browser was apparently not simply window dressing surrounding Chromium, which are pretty much everyone else's web browser, including Microsoft's own Edge. So they explained: "DuckDuckGo for Windows was built with your privacy, security, and ease of use in mind. It's not a 'fork' of any other browser code." All the code...

Leo: Oh, that's interesting. Their own engine.

Steve: Yes, yes. Well, kind of. "All the code, from tab and bookmark management to our new tab page to our password manager, is written by our own engineers. For web page rendering, the browser uses the underlying operating system rendering API."

Leo: Oh.

Steve: "In this case, it's a Windows WebView2 call that utilizes the Blink rendering engine underneath." So that's interesting. On the other hand, what this means is this is all virgin code. And, like, you know, don't trust it very far, right, because Microsoft abandoned Blink in order to switch to Chromium for Edge. So DuckDuckGo has come along and said, okay, we're going to use Blink.

Leo: Is Blink the Internet Explorer engine? I guess it is.

Steve: Well, no, it was the, yeah, the IE11 engine.

Leo: Oh, my god.

Steve: I know.

Leo: Okay. What could possibly go wrong?

Steve: Exactly.

Leo: On Apple I presume it uses WebKit, which is a pretty up-to-date standard.

Steve: Yes, yes, exactly. So they finish by saying: "Our default privacy protections are stronger than what Chrome and most other browsers offer, and our engineers have spent lots of time addressing any privacy issues specific to WebView2, such as ensuring that crash reports are not sent to Microsoft." Because of course the crash report would also tell, like Microsoft, which URL you had pulled which caused their pressure browser rendering engine to crash, and then so they can go fix it.

So, okay, Leo. Since Paul Thurrott appears to have an interest in exploring the experiences and features offered by various web browsers, perhaps when the subject of web browsing next comes up, as it probably will tomorrow...

Leo: Tomorrow, yeah, yeah.

Steve: Yeah, just mention DuckDuckGo for Windows.

Leo: I'll ask him about it, yeah.

Steve: And, yeah, see if he wants to go poke at it.

Leo: He has certainly tried DuckDuckGo on his iPhone and his Macs. So it won't be unfamiliar to him.

Steve: No.

Leo: And then I think many of us use their search engine. So, yeah.

Steve: Yeah, yeah.

Leo: All right.

Steve: While we're on the subject of browsers, I'll note for the benefit of any of our Tor browser users that version 12.5 has just been released. It supports a bunch of UI improvements, including a redesigned visualization of the Tor circuit which shows the Tor onion router hops between you and whatever site you're visiting. Basically, you used to have to go to a separate place in the browser. Now, in the same way that you can click on the URL bar to like show certificates and things, now in the redesigned UI for the Tor browser you're able to click just ahead of the URL, and it drops down a little window showing you a cute little circuit diagram of you at this IP, and then the first router at this IP, the second router in the chain at this IP, the third router at this IP, and then the site where you're visiting. So anyway, it's kind of cool.

And finally, one more browser update. Not long ago, everything was blockchain this and blockchain that. You know? Blockchain was the magic pixie dust that was being sprinkled on everything to make it more better. Today, that role has been taken up by the phrase, which we were talking about at the top of the show...

Leo: AI.

Steve: AI.

Leo: It really is the blockchain of - this is blockchain, it really is.

Steve: Exactly. Exactly. I suppose it shouldn't surprise anyone that every other word in Opera's announcement of their "totally rebuilt from the ground up" all new web browser is "AI." So last Tuesday they posted this: "Hey, Opera fans! Today we're excited to drop the big news that Opera One," which is what they're calling it, "the latest incarnation of the Opera Browser, is here and ready for you to download.

"Here's the scoop," they wrote. "Opera One is your familiar Opera Browser," except as we'll see in a minute it's not. "But it's been given a major makeover. And we're not just talking about a new coat of paint. We've reimagined and rebuilt Opera from the ground

up, paving the way for a new era in which AI is not just an add-on, but a core part of your browsing experience. So what's actually new? Well, for starters, Opera One is introducing Aria, the first-ever native browser AI. There's also a totally fresh Modular Design and a bunch of game-changing features like Tab Islands, ingrained within the browser.

Okay, now, I'm not going to spend any more time on this. And from the comments in the announcement's posting - which was the posting was long, the comments were at least as long. From what could tell, this totally new look, feel, and AI were not going over very well with existing Opera users. And in fairness, big changes always have that risk, right, like this is a completely changed look. It doesn't even look like a browser. It's got super roundness, and things are floating around, whatever these Tab Islands are, you know, unfortunately it may be Gilligan's three-hour tour. It just doesn't look like this thing is going to go. But anyway, for what it's worth, there are Opera fans out there. Wanted to let everyone know, Opera One, take it or leave it, it's got AI in it. I don't know what that means. But if you're curious, you can find out.

Okay. As we've reported, the Kremlin in Russia is now moving away as quickly as possible from Western-made smartphones. And this is like one of those, why did it take them so long? Because, yeah. So it only makes sense that they would turn to their own well-regarded Kaspersky for a solution. To that end, Kaspersky has previewed the first version of their KasperskyOS, a "hack-resistant," we don't know what that means exactly, but good, mobile-targeted operating system that they've been developing for the past several years.

It was demonstrated at a business conference recently held in Saint Petersburg just earlier this month, with the initial version equipped with a bare bones set of basic applications for phone calling, SMS messaging, an address book, and a settings panel. So again, bare bones. Kaspersky says it's currently working on adding a Chromium-based web browser and support for a camera, WiFi, and NFC features. They are looking for a partnership with a hardware smartphone vendor to produce a finished product which will eventually be made available on Russia's internal market.

And I don't have to tell them this because these guys know what they're doing. But if you want security, you need to hold back on features; right? I mean, you're not going to be competing with iOS or Android unless you want to just give up on security. I don't know what hack-resistant means, but keeping this thing to a bare minimum of features is the way to keep it secure. So it'll be interesting to see how this evolves. And this would, of course, provide an answer to Russia's need for something more secure than go buy an Android device from a Chinese vendor, which is what they've been saying up to this point.

And while we're on the subject of Russia, the cost of doing web hosting business in Russia just increased. So I suppose that means that the cost of web hosting to Russian citizens located within Russia will also be increasing as those costs are passed along. Last Thursday, our favorite Russian Internet watchdog, Roskomnadzor, named the 12 largest and most popular Internet hosting companies who must participate in some new legislation. I had Google translate Roskomnadzor's announcement from Russian.

According to the legislation, foreign hosting providers whose users are located, among other locations, on the territory of the Russian Federation, are subject to Federal Law No. 236-FZ, which is titled "On the activities of foreign persons on the Internet in the territory of the Russian Federation." Inclusion in this list of entities imposes obligations on foreign hosting providers to open a branch, a representative office, or some legal Russian entity in Russia, post an electronic feedback form for Russian users on their website, and register an account on the Roskomnadzor website for interaction with local Russian authorities. Failure to comply with the legislation risks the imposition of fines and

even access being blocked to their infrastructure. And the list is pretty much the Who's Who of Internet hosting: AWS, DigitalOcean, GoDaddy, HostGator, DreamHost, Bluehost, Hetzner, WP Engine, Network Solutions, IONOS, FastComet, and Kamatera. So...

Leo: Everyone. Is it?

Steve: Yes, basically everyone. Now, I did notice that Azure is not there. Does Azure do web hosting, or are they just like cloud service stuff?

Leo: Oh, that's a good question. Yeah, you could probably run IIS on Azure and serve it. That's a good question. I don't know.

Steve: Yeah, they were sort of conspicuously missing. Anyway...

Leo: Nor is Google on there, either. Oh, you know why? Because Microsoft and Google both already have offices with humans in them. And by the way, this is the whole point is so that there is somebody they can arrest...

Steve: Yes, exactly.

Leo: ...if they don't like what you're doing, and there's actual collateral damage.

Steve: Yes, yes. Some skin in the game.

Leo: Some skin in the game, literally, yes.

Steve: Yeah. And so these guys are offering their services to Russians inside the Russian Federation without themselves being there. So, yeah.

Leo: No, not going to happen.

Steve: So slowly turn the wheels of justice.

Leo: Oh, yes.

Steve: SolarWinds - remember SolarWinds? Of course we do.

Leo: We do, yes.

Steve: From three years ago. They've said that some of its current and former executives have received what's known as a "Wells notice" from the U.S. Securities and

Exchange Commission for their role of overseer, you know, the SEC is in the role of overseer of publicly traded companies. The notice in this case is in connection with the company's devastating 2020 security incident, which is of course why we all, and the only reason we all, know the name "SolarWinds." A Wells notice is a letter that the SEC sends to companies when the agency is planning to bring an enforcement action against them. SolarWinds says the SEC may fine or bar some executives from serving as officers or directors of public companies. So, you know, you can't completely hide behind the corporate shield, especially when something this bad happens.

Last Friday, the Senate Armed Services Committee announced that it will be formally exploring the idea of creating a new dedicated Cyber Force branch of the U.S. military. So, I mean, it'll be standing alongside the Army, the Navy, the Air Force, the Marine Corps, Coast Guard, National Guard. And of course we have the Space Force. Now looks like we're on our way to having, like, an official Cyber Force as a branch of the armed services. To further this, a provision has been added to the 2024 National Defense Authorization Act calling for an assessment of creating such a dedicated Cyber Force branch.

And now, Leo, I have a picture in the show notes here at the bottom of page five which shows this apparently in action. And what I want to know is why do these photos of U.S. Cyber Defense always show guys with shaved heads - that part I understand.

Leo: And camo.

Steve: They're sitting, yes, exactly, they're sitting in front of their screens and keyboards, dressed up in full camo. You know? And is this an attempt to avoid being seen by the webcam?

Leo: It's actually the opposite of camouflage, if you think about it. They would be much harder to spot if they were wearing business suits and ties.

Steve: Yes.

Leo: It's pretty obvious that there is something going on here.

Steve: And I don't think those outfits are comfortable, are they? I mean, I don't know.

Leo: I have to say I've never served our country, and I've never worn them. I don't know, maybe somebody who is in the service knows. Can you wear, I mean, do you have to wear the camo in all, every - is this your uniform that you wear everywhere?

Steve: And what I don't see is a Post-it note. If they just used a yellow Post-it note over their webcam...

Leo: That keyboard looks pretty good. That's like a good keyboard.

Steve: It does look like a nice keyboard.

Leo: That's a fancy - those are fancy switches.

Steve: Although look at the wire, like, is it stuck up in front of the display? I think the whole photo was a setup, Leo.

Leo: Staged. Hey, you've got to get rid of that Logitech Bluetooth keyboard. Here, use this.

Steve: But definitely dress up in camo because we want to show that you went through boot camp in order to boot your computer.

Leo: Yeah.

Steve: I don't know.

Leo: Yeah.

Steve: Doesn't make any sense to me.

Leo: Yeah. ReverbMike says they're comfortable, these BDUs, and khaki's comfortable. He wore them everywhere. So there you go. There you go.

Steve: Good to know. And I wonder if they actually do wear them, like in these cyber...

Leo: I don't - I feel like they wear black T-shirts that say you know, like death metal bands on them.

Steve: Yes, exactly.

Leo: But I might be wrong.

Steve: Like "Boot You" or whatever.

Leo: Boot me.

Steve: Okay. Just a quick note that Apple has added Passkeys support for logging into Apple.com. You will need to wait for the formal release of iOS 17, iPadOS 17, or macOS Sonoma to be able to do that, or be using a beta. But for what it's worth that support is there now. I suppose that other Passkeys clients should also work now, as well. So if you're looking for somewhere to log in, you can do that at Apple.

Okay, now, here's a bit of sadness that actually we'll be coming back to at the end of the podcast. Several European governments, specifically the French, German, and Dutch officials, are pushing the EU to add an exemption in its upcoming European Media Freedom Act (EMFA) which would explicitly - I can hardly say, I can hardly believe this - explicitly allow EU member states to continue spying on the electronic communications of journalists under the guise of "national security." The push follows the results of the EU's own PEGA commission, which advised the EU to head in the opposite direction by adding additional safeguards to protect democracy and the rule of law in the EU against the abuse of spyware tools.

In PEGA's report published last year, the commission said several EU countries were abusing surveillance technologies to illegally spy on their own citizens, including journalists, under murky and vague "national security" justifications. More than 60 journalistic organizations and civil society groups have signed a joint letter to the EU Council advising against weakening the upcoming law and giving governments an explicit spying carte blanche. So, yeah. Apparently everyone else gets constrained by the GDPR and all that it brings, but the governments themselves which are behind the GDPR are seeking to legislate a loophole to allow themselves to use spyware, which of course is in itself, let's not forget, illegal malicious software. Unbelievable.

Okay, now, it may be obvious to everyone, but I think it's still worth reminding everyone that just because Apple did a beautiful job and got the whole facial recognition challenge correct, that fact should in no way confer any presumption that anyone else did the same. A recent study updated an earlier study from four years ago. Both concluded that with the sole exceptions of Apple and Samsung, the phrase "smartphone facial recognition security" is an oxymoron.

The updated research conducted by a Dutch consumer protection association found that facial recognition systems on most of today's mid- to upper-tier smartphones, which is to say the only smartphones that have any, can be bypassed using a simple two-dimensional photograph. The researchers bypassed facial recognition on 26 different smartphone models by showing photos of the owner to their phones. Only Apple and Samsung devices were found to be secure. Researchers were unable to bypass facial recognition on any of Apple's iPhones, and only one out of 12 Samsung models failed the same test. Fourteen of the 26 smartphones that failed the test were Xiaomi models. Among the failures were Motorola Motos, Nokias, a OnePlus, two OPPOs, and one Samsung Galaxy. That one Samsung was a Samsung Galaxy A04s. And then all the rest were just like all of these Xiaomi phones.

Now, of course we'll all remember, because we were all here on the podcast, when Apple first unveiled their facial recognition. The first thing that naturally occurred to all of us was to wonder how easily their technology could be spoofed. What we learned was that the phone projects a scanning dotted grid, an IR grid, which is viewed by offset cameras to determine whether what's being presented to it matches the model of the 3D face that was created and mapped when the phone's user was first presented to it and deliberately moved around to register themselves and create that map. While that system, which is quite sophisticated, can be spoofed by creating 3D replicas of the user's face, no simple-to-create flat photo will do the job.

So I just wanted to remind everyone that, again, just because Apple went to the extreme measures to create a highly spoof-resistant facial recognition and unlocking technology, no one should assume that anyone else who offers facial recognition unlocking also took the time to get it right. Apparently, no one but Apple and Samsung did. Since getting it wrong is so much easier to do, that's what's typically done. And it seemed to me that the danger is that facial unlocking would have started off with a great reputation of being secure, and that other manufacturers would just be riding Apple's coattails by saying, yeah, we've got it, too. Look, you can look at your phone and unlock it. Well, yes. And

apparently you can show it a photo from the Internet and unlock the phone just as well. So just a caution that maybe you want to actually do that if you have a non-Apple or Samsung phone to see how secure that unlocking really is.

Leo: I wonder if that includes Google? Or does Google not have face ID? I guess Google doesn't.

Steve: I think they're big on thumbprints.

Leo: Yeah, yeah, yeah, yeah.

Steve: Okay. Google, speaking of Google, has committed more than $20 million to the creation of cybersecurity clinics at 20 higher education institutions across the U.S. The clinics will provide free cybersecurity training and hands-on experience for thousands of students. Some Google employees will serve as mentors and trainers at some of the clinics. Google will also provide free scholarships to allow some students to attend its Cybersecurity Certificate program. In part of this announcement, Google said: "These clinics provide free security services in the same way law or medical schools offer free clinics in their communities. They give students the opportunity to learn and improve their skills, while helping to protect critical infrastructure such as hospitals, schools, and energy grids."

Now, this sounds like a great idea, though I'll admit that the cynic in me wonders whether this might not also be a terrific means for recruiting talent from those institutions; you know? Not that there's anything at all wrong with doing so. After all, the reason those students are there is to acquire the knowledge and skills necessary to find gainful employment. So getting a head start with Google might be a way to do that.

Okay. So now, finally, I suppose it was inevitable that the subject of last week's "Massive MOVEit Maelstrom," which was last week's title, I suppose it was inevitable that Progress Software would soon be facing lawsuits because the damage that occurred was astonishing. And, sure enough, at least two federal class action lawsuits have been filed so far in connection with this devastating SQL injection vulnerability which was discovered and widely exploited in their software, which of course we covered in detail last week. The lawsuits allege that it was the company's negligence which led to the breach, thus putting their personal financial data, that is, all of the individuals who are bringing these lawsuits, at risk.

The first suit, which was filed on June 15th in U.S. District Court for the Eastern District of Louisiana, alleges that the vulnerability led to the breach of the state Office of Motor Vehicles, which as far as we know it did. Louisiana said that their Office of Motor Vehicles statewide was completely, all of the personal data was exposed. They announced the breach the same day, warning all, that is, Louisiana state, warning all Louisiana motor vehicle drivers that their names, addresses, dates of birth, driver's license numbers, Social Security numbers, and vehicle registrations, and any other information that they had was likely stolen. You know, pretty much the whole enchilada. About six million records were exposed and likely stolen.

The plaintiff in the first case, Orleans Parish resident Jason Berry, alleges that his personal data was put at risk by the breach. He alleges that the company also failed to promptly notify potential victims of the risk of exposing their personal information. The suit seeks class action status for others impacted by the breach. Now, I'll just note that that's nonsense because he brought the suit the same day that Louisiana announced the

problem. So how could Progress Software have known that this was the case until Louisiana said, yup, we were hit by this. So I don't think this stands much chance of going anywhere. And you and I, Leo, are both not big fans of class actions because mostly that just seems like a way to enrich attorneys.

As we were recording last week's podcast on this topic, the second case was being filed in the U.S. District Court for the District of Massachusetts on behalf of also Louisiana. Three Louisiana residents, Shavonne Diggs and Brady and Christina Bradberry, brought that class. The class exceeds 100 people, and the plaintiffs are seeking upwards of $5 million. Now, is that for the whole class, or individually? That wasn't clear. But this is according to the complaint. The second Massachusetts case alleges that Progress Software failed to adhere to Federal Trade Commission guidelines for data security, failed to protect customer data, and failed to properly monitor its own internal systems.

Okay, except that's not the nature of the breach that occurred. And, you know, I don't have any opinion more or less about this one way or the other, that is, in terms of like from the legal standpoint. One issue may be that the plaintiffs need to be more than just upset over the news of this happening. At this point they may just be chasing ambulances. I suspect that they need to demonstrate that they have been individually and collectively damaged by the breach, and that may not be easy. Remember as we talked about last week, the Cl0p gang, who are Russian extortionists, did say that they wanted nothing to do with government, educational, or police agencies, and that any data obtained from any of them would be immediately deleted. So Louisiana is certainly a government agency, as opposed to a private enterprise. So I hope that Progress Software's attorneys are up to speed on that and maybe saying, look, as far as we know, there's no danger here.

Everyone knows quite well that I have no sympathy whatsoever for anyone who designs web server software in such a way that it feeds any user-provided text to a backend SQL database which stupidly mixes commands and query text into the same text stream. Anyone who is still doing that 25 years after it was first observed to be a really bad idea, and with it being consistently the top vulnerability on OWASP's top 10 list of really bad ideas, is probably going to get what they deserve.

But we don't know in sufficient detail how this happened. Remember that back in November of 2015, when Marriott International acquired Starwood Hotels & Resorts, the Marriott execs didn't know that Starwood's network was hosting some serious security vulnerabilities. And three years later, in September of 2018, that oversight came back to bite them hard. Should Marriott have done an in-depth security verification? Yes. And perhaps they did. We don't know. If vulnerabilities were not extremely difficult to find, they would all be eliminated before software was ever shipped, and the entire bug bounty industry and Pwn2Own competitions would not exist. But the fact that bug bounty hunting can be a profession these days, and Pwn2Own is full of previously unknown vulnerability discoveries, it just demonstrates that these things are hard to find.

So in this case of MOVEit and Progress Software, I don't feel any sense of schadenfreude. This is a tragedy all around where everyone has lost. Our listeners know that I always completely separate mistakes from policies. So my only argument here is that the use of SQL in this way, in any way that opens the door for injection, is a policy decision. It was a mistake that this policy was not implemented perfectly. But if this database architecture policy had not been used at all in the first place, then there would have been no reliance upon the filtering code needing to be perfect. And apparently some imperfections were found and exploited.

So it'll be interesting to see over time what happens with this. You know, lawsuits are unfortunate. We're in an industry where - and Leo, I think it was on one of the other

podcasts I heard somebody lamenting the bizarre fact of the hold harmless clauses in software licensing.

Leo: Oh. Oh, yeah. Yeah, yeah. We've talked about that a lot with Cathy Gellis and others on TWiG, yeah. Yeah, yeah. It's gone, I think, yeah.

Steve: It is an anomaly for this industry.

Leo: Basically and we've all read it, if you ever read the EULAs. We warrant no representation that this software will do anything it's supposed to do. We are not responsible for anything it does wrong. It's your problem if it does it wrong. We're not liable. And it's actually coming up because of self-driving vehicles. That's the latest iteration of this is who's responsible if a self-driving vehicle kills you? Isn't it the maker of the software? And so I think this is going to end up getting rid of the hold harmless clause. President Biden put out a, I don't know, it doesn't have the force of law, but put out a kind of future of technology thing in which they say we don't want these clauses that prevent liability. We want to override them. So I think it's an agenda of the White House, at least, yeah.

Steve: I mean, so it's a problem because it would be difficult to publish software if anyone could sue you if they were not happy with what the software did. And, I mean, there are clauses in there that say our entire liability is to refund the purchase price. Except that, you know, giving you your money back for the car that, you know...

Leo: Yeah, it's not going to do it. Yeah, yeah, that's not going to do it.

Steve: ...that plowed into a crowded group of people, that's not going to work, is it.

Leo: I mean, this is how the court system works now. You can sue anybody for anything. Suing just means I'm going to court.

Steve: Right.

Leo: And the good news is, I would hope in most cases, that judges will throw out frivolous and stupid suits, but maintain suits that have merit.

Steve: And because the attorneys know that, they won't even take up a case when they know that the judge - it was not going to get past square one.

Leo: And there are some states that have SLAPP laws, which I think are probably a good idea, which if it is a frivolous lawsuit and found to be, then the person who brought the lawsuit is liable for costs. And those are effective as a deterrent, as well. But I do agree, and we were talking about this on Sunday with Alex Lindsay, maybe this is what you're remembering, his dad is a trial lawyer. I do agree that that's one of the important ways people can hold these big tech companies accountable is suing them.

Steve: Yeah. So David Scholten, he sent me a tweet. He said: "@SGgrc I have loved listening to Security Now! over the last 10-plus years, and I believe it has helped me greatly in my IT career, from technician to IT admin. Now, I have a non-IT question. Is it just me, or have I been hearing a fire alarm low battery beep in the background in several podcasts?"

Leo: Uh-oh. I haven't heard that.

Steve: David, thank you. I'm glad you haven't, Leo. Many of our listeners have.

Leo: Last week?

Steve: David, I wish it was your imagination. No, it's been going for several months.

Leo: You're kidding. You can't find it?

Steve: I'm not kidding. I cannot find it. Something in my environment started beeping occasionally many weeks ago, and I have no idea what it is or where it is.

Leo: So frustrating.

Steve: It's not any of my smoke detectors. And the room it's in is full of equipment, so there are a great many places it might be. Since it began, I've embarked on several missions to locate and find it.

Leo: Oh, my god.

Steve: But the chirp is so short that I don't get enough of a sample to obtain a bearing.

Leo: And plus it's a high frequency, so it's hard to figure out. You know what? You wrote, years ago as a youth, you did the Portable Dog Killer. I think you need to write something, make something called Chirp Finder.

Steve: A high-frequency sound locator.

Leo: Yeah. Yeah. Yeah. Because it has to record it and do it instantaneously. You could do this, Steve. It's not too late.

Steve: Fortunately, SpinRite is almost finished.

Leo: This can be your next project.

Steve: Yeah, it's...

Leo: Chirp Finder.

Steve: So I tend to tune it out. I really don't hear it that much. But, and I was self-conscious about the podcast, but since the Heil microphone is pointing away from the room where it's happening, I thought, okay, it's probably not going to be very much.

Leo: I remember hearing it several months ago. But I haven't heard it lately.

Steve: It's been going on nonstop. And I'm waiting for the battery to die.

Leo: Oh, my god.

Steve: So far that hasn't happened.

Leo: Oh, my goodness.

Steve: Anyway. And it's funny, too, because since I'm unable to determine the bearing, I'll wait for it to happen. Then I'll go stand over near where I think it is.

Leo: Yes. Everybody knows this look.

Steve: And wait again.

Leo: We're all waiting for that. Oh, I hate that. I hate it. And of course it always happens in the middle of the night; right? There is a great - there's a TV show you probably haven't seen, I think it's on HBO, with the guy who was in "House." I can't remember his name. But he's the captain of a...

Steve: Hugh Laurie.

Leo: Hugh Laurie is the captain of a - it's a science-fiction comedy of a cruise ship, space cruise ship. It's called "Avenue 5." And there's one whole episode devoted to a beep. And it beeps. It doesn't beep consistently. It beeps at random intervals. No one can sleep. Some people laugh every time it beeps. Some people hunch over. It's actually a very funny episode, if you get a chance to see it. I'll find the episode number. Actually, I watched the whole thing. I thought it was pretty good.

Steve: "Avenue 5."

Leo: "Avenue 5."

Steve: I like Hugh Laurie a lot. It's freaky that he has an English accent because none of that shows when he's playing Dr. House.

Leo: Oh, it's a big part of the show. There's a lot of tongue-in-cheek. It's actually very funny. It's the guy who did "Veep." He's very talented. Armando Iannucci. But it just missed slightly, and I guess it was canceled. But it was good. Anyway, look for that.

Steve: I'll track it down.

Leo: You need a Chirp Finder of some kind.

Steve: I know. And I did find myself thinking, how can - somebody must have done this already. Like there must be, like, if you had two phones that were in communication...

Leo: Oh, so you could triangulate it, you mean.

Steve: Yes. You could use time of arrival in order to determine where it was. Anyway. Don't know. I'll eventually find it.

Fabian Santiago said: "I'm still sore for and with you about SQRL vs. Passkeys, et cetera. It does warm my heart to see the SQRL iOS Testflight Client App still receiving updates, though." He said: "Just today for me."

So I wanted to take this opportunity to give Jeff Arthur, SQRL's iOS client author, a shout-out and a thanks for his continuing work on SQRL. I know that it's been a labor of love for him, and it would be terrific if something were to ever come of it. If FIDO2/WebAuthn and Passkeys evolves to require elliptic curve crypto as one of its available crypto suite options, that would immediately enable the use of SQRL-style deterministic, rather than random, private keys. And that would in turn mean that all of the other work that has been done on SQRL to solve all of the other problems that today's Passkeys clients still have, would be immediately available, too. So we'll see how this evolves. All may not be lost. But I'll be on to SpinRite 7 and beyond by that point.

Leo: And of course Chirp Finder. Very important. You've got to get to work on that, Steve. There's a lot of people out there who would appreciate it.

Steve: It's got to be, to have an opportunity to track that down. Okay. So David R. Bunting, he's tweeted me: "Jungle Disk. Do you still recommend it? Thanks, Steve."

Okay. So no. Jungle Disk, for those listeners who haven't been around since the beginning, was one of our very early and very good TNO, as in Trust No One, client-side encrypted cloud storage solutions. They were purchased some time ago by something called "CyberFortress," and those guys probably wear camo, too. And it appears that they've completely gone corporate. So it's unfortunate they got...

Leo: It's not free anymore, and it's - yeah.

Steve: Yeah. They got gobbled up, essentially.

Leo: Such a cool product.

Steve: Yeah, it was. It was great back then. So today my number one favorite choice and recommendation is the Canadian firm and service SYNC.com. You can get 5GB for free to see how you like it, or you can use my referral code to start off with 6GB, so you get an extra gigabyte for free. And that's just grc.sc/sync (S-Y-N-C). So grc.sc/sync. All you have to do is create a username and password, no credit card required or anything else. So it's really absolutely free. Now, I've been using them since, I checked, August 7th of 2019, so we're approaching four years. And in my opinion they are a total win. The only downside is that they don't support Linux. And although they know that there's a demand for it, especially from our listeners, unfortunately the demand for Linux is dwarfed by the interest in Windows and Mac, both which they do support. So there's no sign that Linux is coming.

What I like most about SYNC is that it's probably the right solution for most people because it just works. When it's installed, it creates a SYNC folder in the system's directory tree, and anything that's placed under there is, whether it's folders or files or a complex tree, anything, is kept fully and immediately backed up to the cloud. And it is TNO. It's locally encrypted with all the bells and whistles you would like. You can ask for a link if you want to share a file, and it will be locally decrypted on the recipient's browser. I mean, they really did this correctly.

If you've got multiple machines, all of their SYNC directories are kept fully cross-synchronized through the cloud. And all this is done with deep versioning so that you're able to go back to previous versions from between six months and a year. Using their web interface, you're able to browse back in time to retrieve something, even files you have previously deleted. It's also zero configuration about how often you want to sync. It just syncs everything all the time. Using the Windows tray utility, it's possible to select things you may not want to sync for some reason which are within the tree underneath your SYNC folder, so there's some optional flexibility there.

When I was deep into SpinRite work, all of my code and management scripts assumed that the ASM directory that I use as the root of all my assembly code was at the C: drive's root. But to have it all backed up to the cloud and synchronized between machines, which is what I really wanted because I have two locations, it had to be underneath the SYNC directory. So what I did was I moved the \ASM directory under the SYNC directory. Then I created a Windows NTFS junction link so that an apparent \ASM directory on the root would be aliased to the ASM directory under the SYNC directory.

So nothing needed to change. Everything that's in my code and scripts and everything still referenced \ASM so that everything worked, even though it was actually over in the SYNC directory. And it all worked perfectly. And I have to say there have been several times when the fully automatic detention of previous file versions has come in very handy.

So once again, if you're interested in their free trial, you can use grc.sc/sync, which will bounce you over to them with my affiliate code appended to start you off with an extra gig for a total of six. And then, you know, if you like it, and I just checked, their basic personal plan is $8 a month, which buys you 2TB of storage, and you're able to increase that as needed.

Leo: Did you stop using Syncthing?

Steve: No.

Leo: Okay. Because that would - wouldn't that do what you want to do with your ASM folder?

Steve: It would, although it wouldn't give me the additional level of cloud backup.

Leo: Cloud storage, yeah, yeah.

Steve: Because Syncthing is purely peer to peer.

Leo: Right.

Steve: And in fact I have mentioned that in my show notes. I wanted to mention one other thing. So anyway, so SYNC.com is my current and well-proven, I've been using it for four years, recommendation for a simple-to-use, foolproof cloud storage solution if you don't need Linux clients. Now, I should note, and Leo we've talked about this a little bit, that while I am still using SYNC for many things, I have since switched to using a pair of cross-synchronized Synology NAS boxes. And I am so impressed by Synology. Boy, you know, every contact I have with it I just think, well, these guys got it right.

So I'm using a very nice free Windows utility called @MAX SyncUp, @MAX SyncUp. That synchronizes my various directories on my Windows machine to the local Synology NAS. And then the Synology NASes are - they use whatever they have that is built in in order to mirror each other at my two locations. So basically I've kind of created my own little personal cloud system using two Synology NASes. But I did want to mention to people this @MAX SyncUp because it is a beautiful Windows solution for producing synchronization to local shares. It will also sync to Google Drive. And it's been independently reviewed by a bunch of stuff. You can find them online. And it gets, you know, all the stars.

And before we leave the discussion I wanted to mention Syncthing. I'm glad you made sure I would not forget it, Leo. It is a terrific peer-to-peer cross-platform solution that is quite happy with Linux. I still have Syncthing running on a surviving Drobo, which is Linux-based. And that instance of Syncthing is keeping my wife's fleet of remote Windows laptops synchronized out in the field. And it really is a terrific peer-to-peer solution. And I know, Leo, that you have gotten up to speed on it and like it a lot.

Leo: Well, I've been using it for years. That's really my only backup solution. I put Syncthing on every computer.

Steve: Yup.

Leo: But, and maybe you didn't know this, there is a third-party Syncthing app for Synology. So I make Synology my master Syncthing, and I make sure it's for read-only. It doesn't delete. It only reads. It receives only. So that way you don't have any accidental deletions. It records everything I've ever had. And then I do the same thing as you do with Synology. I have a dual Synology setup, one at home and one here. So that gives me off-site. That gives me my cloud.

Steve: Redundancy.

Leo: Yeah, none of my stuff is ever in a cloud, but it is redundant, and it's offsite, and it's all done with Syncthing. And Syncthing is Windows, Mac, Linux. And that, honestly, I think SYNC.com is great. I just, without a Linux client, it's not going to help me, yeah.

Steve: Right. Do you know if the Syncthing for Synology requires a container? I thought that it was only...

Leo: No. There's two. There are several.

Steve: I didn't think it was native.

Leo: There is a docker Syncthing. But there's a native Syncthing. It's a third-party app, so you have to enable a third-party app store. I think it's from the Syncthing folks. And it runs, not in a - you do obviously have to have some additional software. I think Note has to be running or something. But then it runs, and it does - and so the key with Syncthing, which it took me a while to figure out, is...

Steve: I know. It is funky.

Leo: Well, you have to only let one - start from one place. And that will be the default folder name because it uses a long obscure GUID for each folder name. Let the master, whatever that is, could be your Synology if everything's there, be the name of that, that GUID of that folder; and then make that be the one that introduces to everybody else, and let everybody else say, yeah, I'll take that, I'll take that, I'll take that. My mistake sometimes was creating a documents folder on two different machines. Then you have two documents folders because the GUIDs are different. And so it's better to say this is the canonical documents folder. Let that replicate to everything.

And once I figured that out, it's been working flawlessly ever since. There are a few little things, you know. If it can't copy something, it looks like there's been an error, and it's just, you know, it's just being cautious, just letting you know. And I have seen some people say it's accidentally deleted everything, which it could in theory because any time you synchronize that could happen. That's why it's good to have a read-only copy somewhere. Synology is probably the best one for that. And Syncthing's free, open source. I love Syncthing. It's just amazing, really.

Steve: Agreed. Okay. I thought I'd said I thought I'd said all I had to say about SpinRite.

Leo: It's done. Steve, it's done.

Steve: Well, I caught up with my Twitter feed yesterday and found a very heartwarming pair of tweets from someone whose name is crzy8ers. So crzy8ers tweeted: "Had a catastrophic hard drive failure. All my finished photos were ready for print. Thought my memories were lost forever until someone on Twitter recommended this software, SpinRite by Mr. Steve Gibson @SGgrc." He says: "Here is his website to find SpinRite Data Recovery."

And that tweet included apparently one of his photos, which is beautiful. And then in a follow-up he said: "Don't know how else I can thank you for your amazing Hard Drive Data Recover Software. When I print my next photo book, I'm going to send you a copy. Thank you."

And so he's not a Security Now! follower. He didn't know me from Adam. But someone said, oops, get a copy of SpinRite. And he did, and he ran it, and he got all of his photos back. And so there's that one, and then there's also one of a bear cub up in a tree. And I was thinking, boy, it's dangerous to take pictures of bear cubs, but maybe he had a long lens, and so he was actually in a different state.

Anyway, I'm hoping that SpinRite 6.1's ability to once again run on drives of truly any size, with any format file system, and in a reasonable amount of time, will help to dispel the lingering misperception that SpinRite's day has come and gone. You know, here's fresh proof that SpinRite is still alive and well. And his recovery was done with 6.0. So there's more goodness coming soon. During all of the testing that we've been doing, many of us are watching SpinRite recovering sectors of data just as well today as it ever has, if not perhaps a bit more so, since modern drives have pushed the data storage envelope even further. And I'll just say that I have a few surprises up my sleeve for v7, which is why I'm already committing to that'll be the next thing I work on.

Okay. So three weeks ago, while covering the week's news for Episode 926, which was our "Windows Platform Binary Table" topic, we touched on Kaspersky's discovery earlier in the week of something unknown, which was apparently generating unexpected network traffic, which they had just found crawling around in their network. And the unknown traffic appeared to be originating from some of their iPhones. At the time I quoted them saying: "The malicious toolset does not support persistence, most likely due to the limitations of the OS. The timelines of multiple devices indicate that they may be reinfected after rebooting. The oldest traces of infection that we discovered happened in 2019." Thus four years. "As of the time of writing in June 2023, the attack is ongoing, and the most recent version of the devices successfully targeted is iOS 15.7." Now, that 15.7 turns out to be a clue that we'll get back to at the end.

Okay. So recall that they were examining iPhone backups to detect traces of this infection. And they had named this still-unknown malware campaign "Operation Triangulation." That being the title of today's podcast, you might expect that we're returning to this because they now know a lot more than they did then. And their knowing a lot more coincides with the need all iOS, iPadOS, macOS, and watchOS users had to restart their devices last Wednesday when Apple pushed out a raft of emergency updates in response to what Kaspersky discovered.

Okay. So what did Kaspersky discover? They used mobile device backups to look at partial snapshots of those devices' file systems. And from what they determined, there's this sequence of events: The target iOS device receives a message via the iMessage service with an attachment containing an exploit. Without any user interaction, thus zero-click, the message triggers a vulnerability that leads to code execution. The code

within the exploit downloads several subsequent stages from the command-and-control server, and that includes additional exploits for privilege escalation. After successful exploitation, a final payload is downloaded from the command-and-control server that's a fully-featured APT, Advanced Persistent Threat, platform. The initial message and the exploit in the attachment are deleted.

So they explained that at the network level, a successful exploitation attempt can be identified by a sequence of several HTTPS connection events. They said: "Legitimate network interaction with the iMessage service, usually using the domain names *.ess.apple.com. Then download of the iMessage attachment, using the domain names .icloud-content.com and content.icloud.com." So they're able to see those interactions. Regular non-malware iMessage attachments will do the same thing.

Then: "Multiple connections to the command-and-control domains, usually two different domains." And I'll share a list here in a second. "Typically netflow data for the command-and-control sessions will show network sessions with significant amount of outgoing traffic." So a lot of data flowing out from the phone. And that makes it a little unusual. "The iMessage attachment is encrypted and downloaded over HTTPS. The only implicit indicator that can be used is the amount of downloaded data is about 242Kb." So basically they're reduced, as we can see, to relying upon metadata since they have no visibility into the phone.

Then they said: "Using the forensic artifacts, it was possible to identify the set of domain names used by the exploits and further malicious stages. They can be used to check the DNS logs for historical information, and to identify the devices currently running the malware." That is, you know, so if you look at DNS logs, depending upon how far back you have them, you will spot DNS lookups to these domains which you now are able to associate with this active malware today. And based on which devices you have which are today generating DNS queries to those domains, you now can determine which of your iOS devices are currently infected. So those domains are addatamarket[.]net, backuprabbit[.]com, businessvideonews[.]com, cloudsponcer[.]com, datamarketplace[.]net, mobilegamerstats[.]com, snoweeanalytics[.]com, tagclick-cdn[.]com, topographyupdates[.]com, unlimitedteacup[.]com, virtuallaughing[.]com, web-trackers[.]com, growthtransport[.]com, and then anstv[.]net and ans7tv[.]net.

So they're obviously meant to appear kind of like benign, generic, like if you saw that happening you'd go, okay, you know, web-trackers.com, of course. Tagclick-cdn.com, yeah. And like, okay. Virtuallaughing.com? Well, who knows. But okay. So again, wouldn't raise any red flags necessarily, and especially when you consider all the other, like, you know, remember that a website, when you load it now, has hundreds of other DNS lookups that are occurring. So this would just get lost in the noise.

So essentially they are unable to see into their iOS devices, which they're sitting here like they're holding them, and they know they're infected with malware because they've been able to see what's going on. All they're able to see is the metadata traces of what these devices are doing. And they're able to get additional metadata from examining iPhone backups and from the these DNS lookups that they're able to intercept. So as I noted before, this whole process of iPhone security serves as a double-edged sword. It attempts to prevent malware from gaining a foothold into the device. But it just as strongly prevents legitimate researchers from gaining a foothold to understanding any malware that does manage to get into a device.

And one of the distressing and growing trends we're witnessing is that these incursions are not arising from some black hat bad guys wanting to sneak into our devices. The driving forces here appear to be legitimate democracies - well, and in some cases autocracies, but even democracies such as those in France, Germany, and the Netherlands. And those are the only ones who have raised their hands to ask whether

this could please be made less illegal and unofficially sanctioned. We know that more traditionally repressive regimes are also doing the same without asking for anyone's permission.

So my point is, the more we learn about the increasing pressure to subvert the privacy of our personal communications devices, predominantly coming from the world's governing bodies, the more happy I'm becoming that Apple has been steadfastly working in this direction from the beginning; you know? There was a time, maybe 10 years ago, when all this effort that Apple was putting into this seemed a bit like overkill. Well, I no longer think that.

Unfortunately, we're still talking about this today because they haven't yet succeeded in getting it 100% buttoned down. And it's not even clear that it's going to be possible. While we're still using our current hardware architectures and our current software models, all the evidence suggests that new critical bugs are being introduced at about the same pace as old bugs are being found and eliminated. Windows is certainly showing no signs of running out of bugs to patch; and nor, unfortunately, is iOS. While it's true that iOS may have many fewer of them per month, it only ever takes one.

Okay. So back to Kaspersky. In their pursuit of this malware over the past three weeks, they've posted a series of updates, their most recent one being last Wednesday, coinciding with Apple's release of patches for the zero-day, zero-click problems Kaspersky has uncovered.

So Kaspersky wrote: "Over the years, there have been multiple cases when iOS devices were infected with targeted spyware such as Pegasus, Predator, Reign and others. Often, the process of infecting a device involves launching a chain of different exploits, for example, for escaping the iMessage sandbox while processing a malicious attachment, and for then getting root privileges through a vulnerability in the kernel. Due to this granularity, discovering one exploit in the chain often does not result in retrieving the rest of the chain and obtaining the final spyware payload.

"For example, in 2021, analysis of iTunes backups helped to discover an attachment containing the FORCEDENTRY exploit. However, during post-exploitation, the malicious code downloaded a payload from a remote server that was not accessible at the time of analysis. Consequently, the analysts lost the ability to follow the exploit.

"In researching Operation Triangulation, we set ourselves the goal to retrieve as many parts of the exploitation chain as possible. It took about half a year to accomplish this goal. And after the collection of the chain had been completed, we started an in-depth analysis of the discovered stages. As of now, we have finished analyzing the spyware implant and are ready to share the details."

Their comment about this taking them half a year took me by surprise. I had assumed that when said they had caught this malware in their network, they meant a week or two before. But they apparently meant half a year ago, and that they've only recently been making the results of this ongoing research public. And now in retrospect that does make more sense, since what they were revealing is far more than a week's worth of effort at reverse engineering.

So they said: "The implant, which we dubbed TriangleDB, is deployed after the attackers obtain root privileges on the target iOS device by exploiting a kernel vulnerability." Okay. So what I believe is that they found that kernel vulnerability, told Apple about it, and that's what got fixed. That corresponds with the CSV that I'll be wrapping up with here in a second. It doesn't look like they have yet found the iMessage sandbox escape, nor the transient attachment which is what gets in there, talks to the command-and-control server, and then downloads the final Advanced Persistent Threat.

What they finally got, probably by intercepting the TLS communications, setting up a TLS interception proxy, and then using that in order to decrypt the HTTPS transaction when one of their infected devices reached out to the command-and-control server to download this final piece, they were able to obtain the final piece and then reverse engineer it. And that's what they're now talking about today.

Being able to foreclose the kernel vulnerability might stall this, but it means that we still have the other parts of the attack chain that, as far as we know, they're saying they have not yet been able to obtain.

So they said: "It's deployed in memory, meaning that all traces of the implant are lost when the device gets rebooted." That's what we have known from what they said before. "Therefore, if the victim reboots their device, the attackers have to reinfect it by sending an iMessage with a malicious attachment, thus launching the whole exploitation chain again. In case no reboot occurs, the implant uninstalls itself after 30 days, unless this period is extended by the attackers. The TriangleDB implant is coded using Objective-C, a programming language that preserves names of members and methods assigned by the developer. In the implant's binary, method names are" - yes, yes.

Leo: You don't need a symbol table. You've got them. They're built in.

Steve: Yes. "Method names are not obfuscated; however, names of class members are uninformative acronyms, which makes it difficult to guess their meaning." Okay. So in other words, exactly as you said, Leo, a huge aid to anyone wishing to reverse-engineer Objective-C code. The names, and thus the purpose and intentions, of the code routines remain visible. But in this case the names of the variable parameters they are exchanging are not useful. Examples of method names which they found are populateWithFieldsMacOSOnly, populateWithSysInfo, getCInfoForDump, unmungeHexString, and getBuildArchitecture.

So having those names is far more useful than unnamed hexadecimal address offsets which is all that's generally available from any language that compiles all the way down to native machine code after any space-wasting symbols have been removed. Although the variable names that were contained in the exploit code are far less useful, they noted that in many cases it's possible to guess what their acronym names mean from context. For example, osV is the iOS version, and iME contains the device's IMEI.

Anyway, they continue to explain: "Once the implant launches, it starts communicating with the command-and-control server, using the Protobuf library for exchanging data. The configuration of the implant contains two servers, the primary and the fallback. Normally, the implant uses the primary server, and in case of an error it switches to the fallback server by invoking 'swapLpServerType' method." And again, you're able to see the name of that in the code.

"Additionally, the sent and received messages are encrypted with symmetric (3DES) and asymmetric (RSA) crypto. All messages are exchanged via the HTTPS protocol in POST requests, with the cookie having the key g, and a value that is a digit string from the public KI configuration parameter." So the cookie has some of the public key parameters used for doing the RSA crypto. Basically they've been able to completely reverse-engineer the thing that runs in RAM after this exploit finally is finished getting itself installed into the system.

They said: "The implant periodically sends heartbeat beacons that contain system information, including the implant version, device identifiers (the IMEI, the MEID, the

serial number and so forth), and the configuration of the update daemon, whether automatic downloads and installations of the updates are enabled."

So my first thought upon hearing that was that it was interesting that heartbeat data was being periodically sent, since that makes this thing more noisy and thus more prone to discovery. But then it occurred to me that an iPhone is probably already extremely noisy with all of the legitimate traffic that it has going back and forth. So any heartbeat data, which is relatively infrequent and not that much, you know, not high-bandwidth, is likely able to hide in plain sight without fear of discovery.

They said: "The command-and-control server responds to heartbeat messages with commands. Commands are transferred as" - oh, and I should also mention that one reason you need a heartbeat to be outgoing from the phone is that holds open any NAT that you've got between the outside public Internet and wherever your phone is behind that. So if you didn't have an occasional heartbeat going out, there would be no way for command-and-control to access the phone behind NAT because there would be no mapping. The NAT would look like the one-way valve it is, like a firewall. So having a heartbeat creates an opportunity for commands to be sent back to this implant.

They said: "Commands are transferred as Protobuf messages that have type names starting with CRX. The meaning of these names is obscure. For example, the command listing directories is called CRXShowTables, and changing C2 server addresses is handled by the command CRXConfigDBServer. In total, the implant we analyzed," they said, "has 24 commands designed for" - and they're shortened them down into five categories. "Interacting with the filesystem (creation, modification, exfiltration and removal of files). Second, interacting with processes, listing and terminating them. Third, dumping the victim's keychain items, which can be useful for harvesting victim credentials. Fourth, monitoring the victim's geolocation. And, finally, running additional modules, which are Mach-O executables loaded by the implant. These executables are reflectively loaded, and their binaries stored only in memory."

So their documentation lists each of the individual commands, each of those 24, in details, and explains each one's purpose. I won't enumerate them here, but it should be abundantly clear that essentially this represents a full and deep remote takeover of any exploited iPhone.

Okay. And get a load of this. They said: "One of the interesting commands we discovered is called CRXPollRecords. It monitors changes in folders, looking for modified files that have names matching specified regular expressions. Change monitoring is handled by obtaining a Unix file descriptor of the directory and assigning a vnode event handler to it. Then, whenever the implant gets notified" - it's proactively notified by the file system - "of a change, the event handler searches for modified files which match the regex provided by the attacker." Think for a minute about how sophisticated this thing is. Then such files are scheduled for uploading to the command-and-control server.

So in other words, it's possible for the command-and-control server to prime the Advanced Persistent Threat implant in a device, to autonomously notify the server when something in that device happens of specific interest to it. When a change in the contents of a directory occurs, a check is done for relevancy. And if that comes back affirmative, the files in question are queued for transmission. In a very real sense, it is no longer your iPhone in your pocket. It is theirs. Talk about being pwned.

They said: "While analyzing TriangleDB, we found that the class CRConfig, used to store the implant's configuration, has a method named populateWithFieldsMacOSOnly. This method is not called anywhere in the iOS implant; however, its existence means that macOS devices can also be targeted with a similar implant. The implant requests multiple entitlements, permissions, from the operating system. Some of them are not used in the

code, such as access to camera, microphone, and address book, or interaction with devices via Bluetooth. Thus, functionalities granted by these entitlements may be implemented in modules." Which they hadn't seen.

Then, at the end of the work of assembling all of this, I found an earlier note written by Eugene Kaspersky himself. And this was written at the beginning of this month. He said: "We believe that the main reason for this incident is the proprietary nature of iOS. This operating system is a 'black box' in which spyware like Triangulation can hide for years. Detecting and analyzing such threats is made all the more difficult by Apple's monopoly of research tools, making it a perfect haven for spyware. In other words," he said, "as I have often said, users are given the illusion of security associated with the complete opacity of the system. What actually happens in iOS is unknown to cybersecurity experts, and the absence of news about attacks in no way indicates their being impossible, as we've just seen."

Okay, now, I thought that was very interesting. He's clearly annoyed, and that's a bit of sour grapes, by their inability as security researchers to obtain any visibility into what's going on inside an iPhone. At the same time, you know, they are Russian security researchers, and I've never seen any reason to mistrust them, but there are people who are unhappy that Kaspersky is in Russia. As we've seen, and as he has said, they are limited to monitoring encrypted traffic for metadata, and making iPhone backups, and sifting through that detritus for clues. I can understand his frustration when they are also targets of these attacks.

And what he just said echoes that thought that occurred to me a few weeks ago when I realized that Apple's high level of security has the unintended effect of protecting malware from discovery. He has just said exactly that.

So this is everything that Kaspersky has publicly shared so far. And the glaring piece of information that is lacking, perhaps because it's unknown, is any commentary about how this thing crawls into iPhones by escaping from Apple's security controls. We have one clue about what I think is probably the late stage of this, which are thanks to the CVE which is associated with one of Apple's updates last week. This is CVE-2023-32434 titled "Integer overflow in kernel." Apple wrote: "An app may be able to execute arbitrary code with kernel privileges. Apple is aware of a report that this issue may have been actively exploited against versions of iOS released before iOS 15.7." And then credit is given to three Russians who all work for Kaspersky.

So it appears that Kaspersky knows a little bit more than what they were saying because they didn't talk about that aspect of it for the time being. And given that this vulnerability apparently enables the later stage of a powerful zero-click iPhone takeover, hopefully we'll never learn more because we really don't have to. It'll be patched. But there'll be phones that will never get patched, so it's better that it's just left unsaid.

Oh, and one last piece of information that came from Eugene Kaspersky was an explanation for their choice of the name "Triangulation," which I had been wondering about. He wrote: "P.S.: Why the name Triangulation? To recognize the software and hardware specifications of the attacked system, Triangulation uses Canvas Fingerprinting technology, drawing a yellow triangle in the device's memory."

So what he means there is that it's possible, and often used, to ask graphic rendering software to draw into an offscreen buffer. And it turns out that the precise details of one graphic renderer compared to another may differ ever so slightly. The difference might be invisible to the naked eye. But, for example, when a diagonal line is drawn, as when rendering a triangle, the exact values chosen by the line-smoothing, anti-aliasing algorithm might differ from one generation or model of a device to another. The practice

known as "Canvas Fingerprinting" uses those invisible yet significant details to tell devices apart.

So thanks to Kaspersky's intrepid work, with their forensic analysis being actively impeded every step of the way by the very security they were trying to strengthen, last Wednesday's Apple updates foreclosed upon a kernel vulnerability that had apparently been in active use for at least four years. We'll never know who or why or what or where. But at least now we know how.

Do the bad guys have another way in? Unfortunately, that seems more than likely. What's most annoying and a bit galling, though, is the idea that our own governments may be the customers for whatever comes next.

Leo: Yeah. It's almost certainly nation-states; right?

Steve: Yes, yes.

Leo: Sure these are very expensive.

Steve: Yup. Oh, boy. To be able to purchase that kind of capability, to send anybody you want who has an Phone an iMessage and then have that level of access to their device, you know, to be able to - in fact, there was some mention that, like, any previous video or audio recordings are immediately exfiltrated and sent back.

Leo: Right. Would it be, I mean, I've heard other researchers complain that Apple's security makes it hard for them to, for instance, take a look at any given iPhone and know whether it's compromised. Right?

Steve: Right. Exactly. In fact, here the only way they knew was by looking at the communications traffic from the device.

Leo: Yeah, yeah.

Steve: Because you can't see inside it. It's a black box.

Leo: You know, and I'm thinking about Google's Chromebook, which is also quite secure. And Google keeps that secure by having, you know, a hash of some kind describing the system files. You know, I would wonder, if Apple really wanted to, if there'd be some way that they could show system integrity without revealing the contents of the device.

Steve: I see what you mean. So like, well...

Leo: I mean, Secure Boot works that way with certificates. But it also validates that the firmware is official firmware; right?

Steve: Right. Although the problem is...

Leo: I guess Apple already does that. That's...

Steve: Yeah, I was just going to say, the problem is there's a bug. If there were no bugs, the system would be perfect.

Leo: Right.

Steve: And so it's the imperfection in the security that is the problem.

Leo: Yeah. Okay. That makes sense. Just feel like Apple could make some sort of effort to...

Steve: Be more transparent?

Leo: No, because, I mean, honestly, look, there's security by obscurity, and I don't think that's a good plan. But there is also security by locking the son of a gun down, encrypting everything, and not making it visible to anybody. You know, I think that's fine. It seems like Apple could have some sort of canary or something that would let you know if there had been tampering. Maybe not.

Steve: I think the problem is that the canary can be put to sleep.

Leo: Right. It's a bug, as you said.

Steve: Right.

Leo: So bypasses all security.

Steve: Yeah. You just gas the canary, and then you...

Leo: He's alive.

Steve: Okay.

Leo: He's alive. What are you talking about? He's alive. He's just sleeping.

Steve: Canary only had one foot.

Leo: He's just sleeping, it's okay.

Steve: That's right.

Leo: Do we trust Kaspersky now? I guess in this regard we do.

Steve: I really do. Our listeners know that I hate the broad brush of saying, oh, all Chinese software is bad, and all Russian software is bad. Like all Chinese people are bad. That's just ridiculous.

Leo: Right.

Steve: You know, I think Kaspersky, I mean, they're giving Apple fixes for zero-days.

Leo: Yeah. It's such an interesting conundrum.

Steve: It is, yeah.

Leo: I mean, I probably wouldn't use Kaspersky antivirus. But I think this kind of research is verifiable, so it's not like the Russian - it's not like Putin told him to say this. So I wouldn't put their software on my system. That's a bridge too far, perhaps.

Steve: I have to say I feel the same. Although, well, only because I don't put anyone's software on my system.

Leo: Well, that's right. Can't trust anybody. Steve, you've done it again, a great episode of Security Now!. Thank you for catching me up. I appreciate that.

Steve: Bye.