Flying Trojan Horses

Description: This week, our time-limited quest to answer today's burning questions causes us to wonder, how worried should Android smartphone users be about Google's revelation of serious flaws in Samsung's baseband chips? What great idea should the NPM maintainers steal? What is it that nation-states increasingly want to have both ways? What crazy but perhaps inevitable change is Google telegraphing that it might push on the entire world? Was it possible to cheat at Chess.com, and what did Checkpoint Research discover? What's the most welcome news of the week for the United States infrastructure? And if Trojan Horses could fly, how many propellers would they need? The answers to those puzzles and riddles coming up next on Security Now!.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-915.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-915-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots of topics on the agenda. We're going to kick it off with four extremely serious zero-day flaws in many Android devices. Find out if yours is at risk. Then we'll talk about TikTok, the move to ban it, and some real scary stats and information about the DJI drones from China. All of this coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 915, recorded Tuesday, March 21st, 2023: Flying Trojan Horses.

It's time for Security Now!, the show where we get together and talk with this guy right here, Steve Gibson, all about, well, everything on his mind including security. Hi, Steve.

Steve Gibson: Yo, Leo.

Leo: Welcome.

Steve: Great to be with you again for the Flying Trojan Horses episode.

Leo: We've been working with Midjourney, Stable Diffusion, and others to generate some Flying Trojan Horses. Having some difficulty, oddly enough. But we'll get an image for you.

Steve: Yeah. You got something that's pretty good. But I think we've got a lot of - I know we have a lot of fun things to talk about this week on our time-limited quest to answer today's burning questions. One of them caused us to wonder, how worried should Android smartphone users be about Google's recent revelation of four serious flaws in Samsung's baseband chips? Also, what great idea should the NPM maintainers steal? What is it that nation-states increasingly want to have both ways? What crazy but perhaps inevitable change is Google telegraphing that it might be getting ready to push on the entire world? Was it possible to cheat at Chess.com, and what did Checkpoint researchers discover? What's the most welcome news of the week for the United States infrastructure? And if Trojan horses could fly, how many propellers would they need? The answers to those puzzles and riddles...

Leo: Really. Really.

Steve: ...coming up next on Security Now!.

Leo: You're going to answer that; are you?

Steve: That's right.

Leo: Really. All right. Now to the Picture of the Week.

Steve: So, okay. You know those - what this reminds me of is the old-school railway sidings where it ends like, you know, it's just a stub. And you've got this industrial strength-looking barrier at the end, typically - remember like it had two big pistons that were spring-loaded so that if any cars rolled, tried to roll off the end of the tracks...

Leo: Right, keep them from going off the end, yeah.

Steve: Exactly. It kind of gave it a nice gentle...

Leo: That's what this is, yeah.

Steve: Yes.

Leo: Keep you from going off the end.

Steve: Except for people; right. So we have a sidewalk which stops. And as I guess some municipal code required, in order to make it very clear, maybe it's to keep skateboarders from hitting the bush that is after the sidewalk, there's a sign across it that says "Sidewalk Ends." You know, very much like that other one, we had the gate that said, you know, "Sidewalk Closed."

Leo: Yeah.

Steve: There, there was actually a sidewalk beyond the gate, but it was...

Leo: But it was closed, yeah.

Steve: ...closed for some reason.

Leo: Here there's no sidewalk.

Steve: So of course there was a well-trodden path around that closure.

Leo: And look. There is a trodden path to the left.

Steve: Yes. And that's actually the point. So this sidewalk ends into what looks like - just like a big bush. And one wonders if they didn't feel like tackling the bush, or that the bush came after the end of the sidewalk or before. We really don't know the sequence of events here. But the need for a sidewalk does not end, even though the sidewalk ends.

Leo: That's exactly right. Thank you for observing that. You don't just stop and go back.

Steve: Exactly. What are you going to do? Like, oh, Martha, we have to turn around now. So, no. Instead, clearly, many people have said, oh, look, there's a path to the left of the end of the sidewalk. We'll just continue along that. And sure enough.

Leo: I bet there wasn't for a while, and then they made one.

Steve: Oh, and, boy, this is - now I don't think we're ever going to get a sidewalk.

Leo: No.

Steve: Because the problem has been solved.

Leo: This is a pet peeve of mine. All over Petaluma, because of poor planning and malfeasance in the city council and so forth, there are streets with no sidewalks. I could walk to work easily, except I take my life in my hands because there's no sidewalk for half the route, and you're walking in the street. And it just irks me. It's really - fortunately they're starting to put a sidewalk in out here. Eventually we're going to have a sidewalk. I don't know if it'll go all the way to my house. But I would like to walk to work. It's only a couple of miles. But I don't dare; you know? Put in sidewalks. Let people walk. Anyway, on we go.

Steve: Yes. So one of the more worrisome revelations of the past week came to light last Thursday when Google's Project Zero's standard non-disclosure deadline expired, 90 days after they had informed their Android hardware and software partner, Samsung, of the 18 separate vulnerabilities they had discovered lurking inside Samsung's widely used even by Google's Pixel phones Exynos modems.

And here's the big news: Four of those vulnerabilities are as bad as any can get for an always-connected smartphone. And in fact those four vulnerabilities are so bad that Google has decided to make a rare exception to their standard disclosure policy, you know, which is like fix it by 90 days or else. Well, not "else" in this case. For the sake of the world they are continuing to hold back details because it is that bad.

Leo: Now, is this different than the SS7 baseband exploit? Is this a new - this is a new one.

Steve: Oh, yeah, yeah, yeah. This is not an exploit against the SS7 protocol. These are, as they call them, Internet-to-baseband remote code execution vulnerabilities which require...

Leo: Zero click.

Steve: ...no click. Nothing.

Leo: I just need to know your phone number. That's it.

Steve: That's exactly right.

Leo: Horrible. Horrible.

Steve: So, okay. So they said: "In late 2022 and early 2023, Project Zero reported 18 zero-day vulnerabilities in Exynos Modems produced by Samsung Semiconductor. The four most severe of these 18 vulnerabilities allow for Internet-to-baseband remote code execution. Tests conducted by Project Zero confirm that those four vulnerabilities allow an attacker to remotely compromise a phone at the baseband level" - that is, you know, underneath the operating system, down at the chip, at the cellular modem chip level - "with no user interaction, and require only that the attacker know the victim's phone number.

"With limited additional research and development, we believe," said Google, "that skilled attackers would be able to quickly create an operational exploit to compromise affected devices silently and remotely." In other words, you now know what all of the state-level actors and, for example, what the NSO Group in Israel are busy doing is, like, what what what? Like, you know, come on. Go figure it out.

So they said: "The 14 other related vulnerabilities were not as severe, as they require either a malicious mobile network operator" - in other words, it's got to be carried by the cellular protocol - "or an attacker with local access to the device." Okay. As for where these chips are in use, and thus what devices would be vulnerable attack targets, first of all, none of our listeners, unless you're the King of Siam or something, probably need to

worry; right? This would only be a targeted attack. Script kiddies are unlikely to ever get access to this. But, you know, if you might be a target, then pay attention to these model numbers.

So Google said: "Samsung Semiconductor's advisories provide the list of Exynos chipsets that are affected by these vulnerabilities. Based on information from public websites that map chipsets to devices, affected products likely include: mobile devices from Samsung, including those in the S22, M33, M13, M12, A71, A53, A33, A21s, A13, A12 and A04 series; mobile devices from Vivo, including those in the S16, S15, S6, X70, X60 and X30 series; the Pixel 6 and Pixel 7 series of devices from Google; and any devices that use the Exynos Auto" - as in automobile - "T5123 chipset."

Now, Leo, I'm not hip enough to, like, these model numbers. Are these current smartphones, these series? Do you know, like, how recent these things are? Not clear to me. They said: "We expect that patch..."

Leo: Hold on. I was over here. I was painting the ceiling. Wait a minute. Hold on, I'm back. Yes, I think it is. The S23 is the current Samsung phone, and I believe that has that Exynos chip in it.

Steve: Okay.

Leo: And, you know, Google in its Pixel phones, this is the Pixel 7.

Steve: Okay. So 6 and 7, they both do.

Leo: Yeah.

Steve: Google knows that.

Leo: They have Samsung chips.

Steve: Okay.

Leo: They never said Exynos. In fact, they didn't want to say Samsung. But then we found out, so now we know, yeah.

Steve: Okay. Okay. Sorry to make you run.

Leo: A lot of modern smartphones have Qualcomm.

Steve: Okay, right.

Leo: But these don't. And the Samsungs don't.

Steve: Of course Samsung's not going to use Qualcomm.

Leo: Yeah, no, they're going to use Exynos, yeah.

Steve: Right. So they said, Google said: "We expect that patch timelines will vary per manufacturer." They said: "For example, affected Pixel devices received fixes for all four of the severe Internet-to-baseband remote code execution vulnerabilities in the March 2023 security update."

Leo: Oh, yeah, which came out about a minute ago. Okay, fine. Okay, thank you.

Steve: Yeah. And but this also says, you know, one of the things that we've said is, if you care about your Android smartphone security, then you really want to be with someone who is going to be patching responsibly. And that's going to be Google and Samsung.

Leo: Yup, there it is, the March update. This is it.

Steve: Yup, yup. They said: "In the meantime, users with affected devices can protect themselves from the baseband remote code execution vulnerabilities mentioned in this post by turning off WiFi calling and Voice-over-LTE data in their device settings." So there is a workaround, again. So if you only leave old-school cellular connectivity on, then you're safe. It is the Internet and the data connectivity that is where the vulnerability comes from. And they said: "As always, we encourage end users to update their devices as soon as possible to ensure that they are running the latest builds that fix both disclosed and undisclosed security vulnerabilities."

Leo: See, I get it now because I'm looking at the devices, and they're all in the last couple of years.

Steve: Yeah.

Leo: And that's because VoLTE and WiFi calling is in the last couple of years. So it's clearly in that part of it because, for instance, the S23, the newest Samsung, is not on the list. Which means they figured it out and fixed it. But all of the stuff from the last couple of years, including the Pixel 6 and 7 are.

Steve: Yeah, or maybe the 23 is not because they're using, you know, certainly Samsung didn't know of the problem and fix it in the 23.

Leo: Right, they must be using a different chipset.

Steve: Yes. Or they just reengineered it, and the reengineering didn't have the problem.

Leo: Accidentally fixed it, yeah. They fixed it by accident, yeah.

Steve: Exactly, yeah. So, okay. So here's how Google has positioned their unusual decision not to fully disclose after 90 days, which is a violation of, you know, it's been a hard-and-fast rule for them. So they said: "Under our standard disclosure policy, Project Zero discloses security vulnerabilities to the public a set time after reporting them to a software or hardware vendor. In some rare cases where we have assessed attackers would benefit significantly more than defenders if a vulnerability was disclosed, we have made an exception to our policy and delayed disclosure of that vulnerability.

"Due to a very rare combination of level of access these vulnerabilities provide and the speed with which we believe a reliable operational exploit could be crafted, we've decided to make a policy exception to delay disclosure for the four vulnerabilities that allow for Internet-to-baseband remote code execution. We'll continue our history of transparency by publicly sharing disclosure policy exceptions and will add these issues to that list once they are all disclosed." In other words, they're saying we're not going to tell you what's wrong, but we're at least going to tell you that we're not going to tell you. So, you know, you know that there's something wrong even if you don't yet know what.

So they said: "Of the remaining 14 vulnerabilities, we're disclosing four vulnerabilities that have exceeded Project Zero's standard 90-day deadline today." They said: "These issues have been publicly disclosed in our issue tracker, as they do not meet the high standard to be withheld from disclosure." As the four bad ones do. They said: "The remaining 10 vulnerabilities of those 14 in this set have not yet hit their 90-day deadline" - remember there were some that were disclosed only in 2023, so earlier this year - "but will be publicly disclosed at that point if they remain unfixed."

So the concern here is that this is, these four, this is a big juicy set of very, I mean, like, infinitely, essentially, powerful exploits which every very powerful and really bad actor in the world knows now not only exists, but also roughly where it exists. And as always, we know that there's a huge difference between a patch being available, and that patch being applied everywhere it's needed. Google's Pixel devices, as we noted, were early recipients of those patches; and, presumably, Samsung's devices will be, too. But what about those Vivo phones, and the autos, automobiles that incorporate those chips?

Leo: Those will never be updated, of course.

Steve: Exactly. So, you know, as I said, random people, like most of us, almost certainly have little to fear since script kiddies are never going to get their hands on these. But this is the sort of vulnerability which is exactly what the likes of Israel's NSO Group is looking to add to their Pegasus smartphone spyware, as are other less public state-level actors. So I would imagine that, without exception, the victims of the exploitation of these vulnerabilities would only be those who are highly targeted and valuable. And of course it will turn their phone into listening devices; right? I mean, they'll suck out their messaging history and what they're doing and tracking, and also probably turn the microphone on in order to eavesdrop in real-time.

So anyway, it's no comfort if you might be such a target; but hopefully you've got, you know, you're not using some random also-ran Android device that might have this chip. You've got a Pixel or a Samsung, and it'll get fixed quickly. And as Google noted, there is a way that you could, until your device is patched, you could make yourself secure if you were someone who should be concerned about that. So of course always good to keep our devices patched. But I doubt that most of us have anything to worry about. And I don't since I am an iOS device user.

So, okay. I saw an interesting idea for wrapping the potentially hazardous NPM command within a protective shell. NPM of course stands for the Node Package Manager, you know, "node" as in Node.js (JavaScript). It's the command-line interface to the most popular JavaScript code repository. The idea for this protective wrapper comes from a company named "Socket," who says of themselves, they said: "Secure your supply chain. Ship with confidence. Socket fights vulnerabilities and provides visibility, defense-in-depth, and proactive supply chain protection for JavaScript and Python dependencies."

Now, they call their latest innovation "safe npm." And I'm going to share a bit of their sales pitch, not because I necessarily think that our listeners should go get it, but because it nicely describes the open-source package distribution risks that we've been covering now for quite a while. So Socket explains. They said: "Socket is proud to introduce an exciting new tool, 'safe npm,' that protects developers whenever they use npm install," which is the command that you issue at the command prompt to install some new package into your system. They said: "Socket's 'safe npm' CLI (Command Line Interface) tool transparently wraps the npm command and protects developers from malware, typosquats, install scripts, protestware, telemetry, and more, 11 issues in all.

"Today when you run npm install it's difficult to know which transitive packages will get installed, whether those packages will execute install scripts, or if those packages have been compromised by malware. The average npm package has 79 transitive dependencies. That means installing a single package with npm install will, on average, install 80 total packages." They say: "It's hard, if not impossible, for a developer to audit, let alone even understand, the full list of packages that will be installed. Most of us just cross our fingers and hope for the best.

"Worryingly, any of these 80 packages can declare an install script, third-party shell code, that npm will automatically execute during installation. While there are legitimate use cases for install scripts, it's also a favorite feature of malware authors; 94% of malicious packages used at least one install script. Developers also face the ever-present risk of typosquatting attacks, where an attacker publishes a package with a name similar to a more popular package. It's way too easy for a busy developer to make a typo when running npm install and install the wrong package. Sometimes, however, typos can have disastrous consequences, such as in the case of running npm install webb3 [with two b's] instead of npm install web3."

Anyway, then they show an example of something quite malicious hiding inside that "double b" version of the webb3 package, just, you know, a typo, webb3. Somebody stuck it there hoping that somebody would type it by mistake. And people did occasionally. So they show that. Then they go on to explain. They said: "This type of malware is all too common. Socket has helped to remove over 200 packages for security reasons - malware, ransomware, spam, et cetera - in the past 30 days alone." They said: "To help you get a sense of the scale of the problem, we freely share samples of recently removed npm packages with the public, for non-commercial research purposes.

"In conversations with developers, we kept hearing the same request. Developers want a way to securely and confidently run npm install without the fear of malware or rogue scripts infecting their systems. Our most popular product, Socket for GitHub, already proactively scans GitHub pull requests for software supply chain risks including typosquats, install scripts, and more than 70 customizable issues. But until today, we have not had a good way to protect the developer's local machine from bad packages. That's why we're super excited to share this initial release of 'safe npm' with you today. Socket is proud to introduce a new feature, 'safe npm,'" blah blah blah. And basically they then repeat what they said.

"When a developer attempts to install a malicious or risky package, Socket pauses the installation and informs the developer about the risk, if any are detected. The developer

is given the option to stop the installation and protect their machine before the package is executed or even written to disk. Alternatively, the developer is also free to proceed and accept the risks."

Okay. Now, the reason I'm bringing all this up and wanted to cover, first of all, every single one of those problems with npm we've talked about on the podcast. You know, scripts being run and so forth. So I'm bringing this up, not to take anything away from these guys, but they want $10 per month per user for this, which, you know, if you were only occasionally doing this, seems excessive to me. But more than that, all of this sounds like something that the maintainers of NPM, that package itself and all other similar package managers, ought to have already built into their basic command-line offerings. I have no way to directly influence that happening. But it may be that Bitwarden now supports the superior Argon2 PBKDF thanks to our talking about it here, followed by some of our listeners suggesting it and implementing and pushing it across the finish line.

So if any of our listeners are able to plant a bug in the ears of the guys who are responsible for evolving npm and the other major package managers, I think it's clear that it's well past time for the industry's various package managers to get proactive about protecting their users from all of this nonsense that's going on in the major repositories that they are, after all, pulling their packages from. There's no sign of this abuse calming down. It's not like it's a passing fad. All of the indications continue to be that it's still ramping up.

Since it's the package manager that goes out and retrieves the package on behalf of its user and then follows all of the dependency linkages and makes sure that everything else that's necessary is there, that function, that exact function needs to evolve beyond being what it originally was a few years ago before this all began to happen with the repositories, you know, a simple, no responsibility being taken, trivial command-line retrieval and installation tool. It needs to shoulder, those package managers now need to shoulder a lot more responsibility.

So anyway, when I saw this announcement from Socket I thought, you know, nice that these guys are doing that. You can get that right now from them. And depending upon your level of risk tolerance, you may choose to do that. But what we really need is for this to be made universal, and the package managers need to step up and start taking responsibility.

Okay. So things are getting interesting as an increasing number of governments are looking at their newly strengthened privacy laws and realizing that the behavior of the big tech giants is in contravention of those statutes. We've already been covering some of these events as they've been happening, but here are a few interesting pieces that we haven't talked about before.

Last year, South Korea's privacy watchdog, known as PIPC the Personal Information Protection Commission imposed a pair of stiff fines on Google and Meta for breaking the country's privacy laws by not obtaining - oh, boy - lawful consent from users and tracking their online activity for advertising purposes. The PIPC imposed a 69 billion won - that's still pretty significant - $52 million fine on Google; and a 31 billion won, which is $23 million, fine on Meta. And, you know, both of those giants could have trivially paid the fines. But that would have set a dangerous precedent, and they would also likely have been required to stop doing what they had been fined over, which both companies appear to be certain is required for their businesses to thrive.

So rather than pay up, both Google and Meta have instead elected to countersue the PIPC. In their recently filed lawsuits, both companies argue that it's the website operators who should be responsible for obtaining individual user consent, not their

platforms, which they contend only receive and aggregate this data which is being collected by visitors to the websites. Okay. So there's one piece.

Meanwhile, over in the never-dull European Union, nearly three years ago, back in July of 2020, the CJEU - the CJ stands for Court of Justice for the EU - ruled that a transfer of data to U.S. providers violate the rules on international data transfers which are spelled out in the GDPR. So the CJEU consequently annulled the existing transfer deal "Privacy Shield." This followed their previous annulment of the "Safe Harbor" agreement back in 2015. So while all of this sent shock waves through the tech industry, U.S. providers and EU data exporters just largely ignored the case. Meta's Facebook, like Microsoft, Google and Amazon, has relied on the so-called "Standard Contract Clauses" and "supplementary measures" to continue data transfers and calm its European business partners.

So back in August the consumer protection agency NOYB filed 101 complaints against specific individual websites which were still using Google Analytics and Facebook Tracking tools despite clear court rules making that use unlawful. I mean, it's unlawful to do that, and everyone's continuing to do it. So this NOYB consumer protection group said, okay, let's start turning up the heat here. And now we're talking about this because last Thursday Austria's Data Protection Authority, which is the DSB, has ruled that Facebook's use of its tracking pixel directly violates the GDPR.

A guy named Max Schrems, the chairman of this NOYB.eu, said: "Facebook has pretended that its commercial customers can continue to use its technology, despite two Court of Justice judgments saying the opposite. Now, the first regulator told a customer that the use of Facebook tracking technology is illegal." Oh, and I suppose it's not surprising that also the use of "Login with Facebook" is also illegal since, as we've noted, it's essentially a tracking technology, too. The use of Google Analytics falls under the same regulation and has already been ruled unlawful. The concern is that if any of these tools are used, data are inevitably transferred to the U.S., where the EU claims to be worrying that the data is at risk of intelligence surveillance. They quote, you know, FISA and blanket NSA rights to look at anything that they want to.

So, as I was thinking about this, what strikes me as more than a little ironic is that these governments who don't want their citizens' web pages to contain tracking pixels, or to use U.S.-based services that might send data outside of their Union, nor for their citizens to be using apps with ties to potentially hostile governments, are the same governments who are increasingly up in arms over their inability to intercept their own citizens' end-to-end-encrypted communications, not only when they might deem it necessary through a wiretap-style search warrant, but also in the form of continuous background surveillance monitoring of all visual and written communications for anything that they might deem to be illegal or suspicious. And of course tracking their locations is part of that deal since it doesn't do any good to know what's going on if you're unable to go grab the perpetrators.

So it's apparently okay for the governments to spy on and track their own citizens, but no one else should be able to. They're all about the rights of their own citizens, except when it's they who are violating them. And that's why I named this little piece of news "It's only okay when we do it." So we've got some more news to talk about; but, Leo, I think we should take our second break.

Leo: You bet. Yeah, Max Schrems actually has the laws named after him because he's such an active advocate of this kind of privacy. They call it "Schrems."

Steve: And, you know, it's going to be up to legislators to figure out, like, what happened.

Leo: Yeah. Schrems was chiefly responsible for making sure data for a citizen of a country is stored in that country, not some other third-party country. I think that's a fairly reasonable ask.

Steve: Yup. And I think that's probably what's going to happen is Meta and Google are not going to pull up stakes. They're just going to move their stuff over there.

Leo: Just have a Networks Operation Center over there. It's no big deal.

Steve: Right. So while we're talking about nervous governments, I'll just note that New Zealand put a ban on the use of TikTok by their lawmakers and other Parliament workers. And this ban goes into effect actually at the end of next week, as March ends. And the Scottish government hasn't quite gotten there yet. But officials were "strongly advised" to remove the TikTok app from all their government devices.

Meanwhile, the Australian government has published a lengthy 113-page report it received from academics as part of its own TikTok investigation. The document describes TikTok's deep ties to the Chinese Communist Party. And basically it wasn't clear there was any news there, but they just wanted their own in order to support their own plans. It's viewed as preparation for a government-wide ban that may arrive shortly, that's expected in the next couple weeks.

And of course over here in the states, the FBI and the U.S. Justice Department have launched an official investigation into ByteDance, TikTok's parent, for using the TikTok application to spy on American journalists. And, you know, this is that old news that some rogue employees were, and it turns out apparently indeed, misusing TikTok to spy on one of Forbes' reporters in an attempt to identify that reporter's sources. And ByteDance said that they fired the individuals who surveilled the journalists. So, you know, more of this drum beat. And Leo, I know that you and your two co-hosts on Sunday talked about TikTok. I wasn't able to listen to that, but you said you were going to. Is there anything else that has happened?

Leo: Well, I mean, yeah. It's imminent that they're going to ban it, I think.

Steve: So you think in this country.

Leo: In the U.S., yeah.

Steve: So not just governments, but everybody.

Leo: Yeah. So apparently TikTok says that the Biden administration a couple of weeks ago told them sell it or we're banning you. Whether they'll be able to sell it is a question because the Chinese government has to approve it, and it seems unlikely that they will. They've said in the past, back when Trump tried this, we're not going to sell it because there's technologies that we don't want anybody outside China to

have, AI technologies. In which case that's going to put the Biden administration up against the wall. And I guess they'll have to ban it. They haven't announced that publicly. It was TikTok that said so. The CEO of TikTok is testifying in front of Congress tomorrow or Thursday.

Steve: I think it's tomorrow, yeah.

Leo: And apparently they've paid a bunch of influencers to come into Washington to tell Congress don't ban TikTok. I mean, I kind of have a sympathy for that. My son got his career start on TikTok. It was a huge launching pad for him.

Steve: Well, and can you imagine, I mean, this would be unprecedented, where an app that is this popular literally, I mean, it would go dark; right?

Leo: That's my biggest concern is it's just a bad precedent for the American government to ban an app. You know, there'll be retaliation. American apps will be banned, you know, there are already many of them banned in China, but elsewhere perhaps. And I just think it sets a bad precedent. I understand the security concerns, and I don't think any government person should have TikTok on their phone. They probably shouldn't have smartphones at all.

Steve: And every time we talk about Russia doing one of these bans, we roll our eyes. It's like, oh, boy, you know, repressive regime.

Leo: Yeah, yeah. Welcome, yeah.

Steve: Well, here we come.

Leo: So I just don't know, I don't - absent solid proof that China is doing something with TikTok...

Steve: Right, right.

Leo: I understand the reason you might want to be afeared. And certainly I think it's well within the rights of governments and agencies and the Defense Department and so forth to ban TikTok on those government phones. That's fine.

Steve: Yeah.

Leo: But there are so many people, millions of people, of creators all over the United States who make their living through TikTok. So I have some concerns over that.

Steve: Wow, wow.

Leo: Yeah. I don't know what the answer is. Because I understand the security concerns. I really do.

Steve: Yeah. Okay. So once upon a time, when I was just a wee lad...

Leo: Oh, a hundred years ago, yes.

Steve: I know. It's been a while. I had hair. You could purchase a certificate that would last longer than an all-day sucker. Actually, it would last for a full five years. Those were the days. In fact, those certificates lasted so long that many companies would completely forget all about them until they were surprised when connections to their web servers suddenly began to fail.

Leo: Yes.

Steve: Then it would be a mad scramble to remember, how do we create a Certificate Signing Request again? I don't remember. And, you know, the guy who did that last time, well, he hasn't been with us for a few years. So we need to refigure out the magic incantations that are required. So it was often a lot of excitement, about every five years, give or take.

Well, as we have chronicled on this podcast, since the days when I was a wee lad at the beginning of this podcast, over the years certificates have largely done their job, but we've also had a lot of fun here on the podcast examining the myriad ways they have fallen short, through no fault of their own. One big topic for us was the whole mess of certificate revocation. That was a lot of fun. At one point, our longtime listeners will recall, I created and then immediately revoked my own certificate to demonstrate just how totally broken the Chrome browser's certificate revocation system was. It didn't actually have one.

Chrome happily honored my revoked certificate that other properly functioning browsers knew better, and they blocked it. This, then, forced Google to manually add an exception for my deliberately revoked certificate to Chrome's short list of known bad certificates, even though Chrome still remained blissfully unaware of all other revoked certificates in the whole world due to the fact that its revocation system, as I said, never actually worked. After they did that, when I created another revoked certificate to demonstrate that they had special-cased my first certificate by manually adding it to that short list, well, they decided to just ignore me since I was annoying them, and I'd proven my point.

But, almost inevitably, certificate expiration durations have been creeping downward. They first dropped from their original "set it and forget it" duration of five years, down to three years. Then they dropped to two years. And now we're all at just one year plus one month. And while this is admittedly five times the work as it was when certificates lasted five years, because now they only last about one year, the people responsible for keeping certificates from expiring now tend to always have that in the back of their minds. I know, for example, that GRC's cert will reach its end of life at the end of July this year. So that's not far off. It's not like it's five years from now. So oh, yeah, you know, will I even still be worrying about this then.

So the story behind how the industry's certificate life was cut in half, from two years to just one year, is relevant because a more extreme version of it might be in our not-too-distant future. Recall that three years ago, back in 2020, it was Apple who made the

unilateral decision to stop supporting any certificate whose data of issuance was more than a year and a month earlier than its date of expiration. So for whatever reason, 365 plus 33 is 398 days. And 398 is the maximum distance you can have from "not valid before" to "not valid after" dates, which is what the certificates contain. So since Apple's decree would cause any and all iOS and macOS devices to reject any then non-compliant websites, the rest of the certificate issuing industry had no choice other than to drop their certificate lifetimes to what Apple was now going to require.

But now there's some scuttlebutt that Google, with their ability to also unilaterally control what most of the web does through the operation of their Chrome browser, that they may be considering doing something similar. But Google is talking about reducing certificate lifetime to just 90 days.

Leo: Oh, boy. That's a problem.

Steve: So initially playing nice, Google says that it plans to make a proposal of this to the CA/Browser Forum, right, the CAB, the CA/Browser Forum, which we've spoken of often back in the day when all this was happening more regularly. This CAB is an informal group of browser vendors and Certificate Authorities who meet regularly to discuss industry-wide initiatives and keeping everything on the same track, what fields certificates should have and so forth.

Okay, now, no one expects administrators of every server on the planet to be manually generating and freshly installing TLS certificates every three months. So the point, I mean, the explicit point of Google's recently telegraphed move is to move the entire industry to enforced certificate automation. ACME is the Automated Certificate Management Environment. As we know, it debuted with the free certificate provider Let's Encrypt.

But I know, for example, that my chosen certificate provider DigiCert now also supports ACME automation. And there's a nice ACME client for Windows which will be able to automate the process for my non-Unix servers. So it'll be a matter of maintaining an account, in my case, and a balance with DigiCert, or some means for them to pull money as needed. Then my various servers will be able to serve their own 90-day certificates and notify me only when there's some problem.

In Google's document proposing this certificate lifetime shortening - and it's a polite proposal, right, because, I mean, the presumption is this is going to happen - Google said the following in support of the move to automate certificate issuance. They wrote: "The Automatic Certificate Management Environment (ACME, RFC 8555) seamlessly allows for server authentication certificate request, issuance, installation, and ongoing renewal across many web server implementations with an extensive set of well-documented client options spanning multiple languages and platforms. Unlike proprietary implementations used to achieve automation goals, ACME is open and benefits from continued innovation and enhancements from a robust set of ecosystem participants.

"Although ACME is not the first method of automating certificate issuance and management [and then they cite] CMP, EST, CMC, and SCEP, which all predated it," they said, "it has quickly become the most widely used. Today, over 50% of the certificates issued globally for the Web Public Key Infrastructure rely on ACME." Fifty percent. "Furthermore," they said, "approximately 95% of the certificates issued by the Web PKI today are issued by a CA owner who has some form of existing ACME implementation available for customers." In other words, you won't have - no one's going to have to change CAs. All of the CAs, 95% of them, already support ACME. So all you have to do is ask for certs that way instead of doing it through the web interface.

They said: "A recent survey performed by the Chrome Root Program indicated that most of these CA owners report increasing customer demand for ACME services, with not a single respondent expressing decreased demand." And this means that, before long, ACME support will become a standard feature of any server that needs to support TLS connections, as most do and will.

And in an interesting bit of coming full circle, the reason I tied this back into certificate revocation is that, with Google reducing certificate lifetimes to just 90 days, the fact that their premium flagship web browser does not and never has properly supported certificate revocation becomes much less of an issue since a stolen certificate would, on average, only be useful for half that period of time, about six weeks, before its short life came to an end.

And just to be clear, there is no timetable for any this. But it does appear to be a thing, and it would likely behoove anyone who is now in the process of setting up any new server environment to plan to implement ACME automation sooner rather than later, maybe from the get-go.

Leo: Yeah.

Steve: I certainly would in my case because the change does make sense, and the writing does appear to be on the wall for this.

Leo: Yeah, I don't mind if it's automated. I mean, right now we're going through this hassle. We've got a three-year paid certificate, but every year we have to update it, and we just went through that with a bunch of servers. If we can implement ACME everywhere, I mean, I use Let's Encrypt, and that's three months, and it's automatic, and it's fine.

Steve: Yup.

Leo: So I guess if we can implement ACME everywhere - hmm. I mean, there are definitely going to be people who know how to get a new certificate with a CSR and all that who don't know how to set up a script like ACME.

Steve: Yeah. I'm sure this is going to cause some pain. And Google's point is, I mean, they recognize that they're pushing the world. They're taking the position that, from what we observed on the podcast, security changes move very slowly. If we can get ACME in place such that certs only have a 90-day life, then we will similarly be able to roll any other changes in certificates, like more use of elliptic curve certs, or we decide we want to change to post-quantum certificates.

Leo: Yeah, yes. That'd make it much easier, yeah.

Steve: Oh, my god, it'll be automatic, essentially. So again, you know, we always see that these sorts of changes are difficult to force down everyone's throat. But at some point it makes sense to do that.

Leo: Yeah.

Steve: Google is going to be the bad guy this time.

Leo: Yeah.

Steve: The bad cop. Leo?

Leo: Yes?

Steve: You'll be glad to know that chess is safe.

Leo: I actually know a little bit about this, but I'd like to hear what the story is, yeah.

Steve: So in their blog posting titled "Checkmate: Checkpoint Research exposes security vulnerabilities on Chess.com," they describe how they discovered, reported, and helped fix vulnerabilities in the popular Chess.com platform. Now, for those who don't know, Chess.com is the world's leading platform for online chess games, with over 100 million members and more than 17 million games played per day. I thought that ratio was interesting: 100 million members, 17 million games per day.

So it functions as an Internet chess server, a news website, and a social networking platform with a strong focus on community-based forums and blogs which allow players to connect with each other, socialize, share thoughts and experiences, and learn from each other about playing chess. Chess.com also conducts global championships, which consist of prize money to the tune of a million dollars for the winner and the coveted Chess.com Global Champion title.

So Checkpoint decided to take a close look into the functioning of Chess.com because there had been some allegations of cheating in the past. What did they find? They found a number of ways that the communications with the site could be manipulated to cheat. They discovered that it was possible to win by decreasing the opponent's time and winning the game over time, without the opponent noticing what had happened. They also discovered that it was possible to extract successful chess moves to solve online puzzle challenges and win puzzle ratings. To do this, they intercepted the communications between the client side (the player) and the server (the Chess.com website). What they discovered was that the server was accidentally sending the correct solution to the puzzle.

Leo: Oh, I could use that.

Steve: To the client's side. And that allowed a cheating client to abuse and cheat on puzzle championships, in which the winner gets prize money, by simply submitting the correct moves that the server was inadvertently providing. And also it was possible to modify, in that case also, the elapsed time it took to consider the solution. And finally, they discovered that in communication between two friends on the platform, after approving the friends' requests to connect, an attacker, or somebody taking an attacker

role, is able to intercept the request with a proxy tool and succeed in both manipulating game timing, which allows a quick win, and in solving a puzzle, which raises his score and the value on the platform. So the good news is, today, thanks to Checkpoint's work, the game of chess as played at Chess.com is safer and fairer than ever before.

Leo: I like the "-er" because in fact most Chess.com cheating and most online cheating in chess has nothing to do with a hack.

Steve: Right.

Leo: It just has to do with me having my Stockfish chess game running at the same time on my phone as I'm on Chess.com and entering the moves in.

Steve: Yup.

Leo: And that, unfortunately, is not a hack. That's a little harder to fight. You see, it's fascinating to see how they fight it because...

Steve: There is deep learning at Chess.com.

Leo: Yeah. Well, so the way you do it is chess computers calculate the current balance of the game in centipawns, a hundredth of a pawn; right? And among two human players the lead change is much more variable in centipawns, you know, might be 100 or 200 centipawns and then up and down and up and down.

Steve: Ah, over time.

Leo: But when a machine plays, it's a fairly linear gain of centipawns. You don't have the same variations. And so it is a little easier to spot a machine player because it doesn't make the same mistakes that humans do. Humans blunder. And so a machine never blunders. And then there's other ways to tell. They play lines that are kind of more machine-like, although that's getting harder and harder because as they get better and better, they look more and more like humans. But really they play too well. That's the easiest way to spot them.

Steve: Right.

Leo: And for instance there's been a big scandal, a super grandmaster named Hans Niemann has been accused of cheating because his rating went up, you know, went up normally, fairly steadily, and suddenly soared in a very unusual way. And furthermore, his results in games where he's over the board and there are measures taken to prevent cheating, his results are not nearly as good.

Steve: And he was actually playing against Magnus; wasn't he?

Leo: Magnus lost a game to him, which is very rare. Magnus is easily the best player in the world. Lost a game to him and then - you can't accuse somebody of cheating. That's not allowed. So he intimated...

Steve: And not very sportsmanlike.

Leo: Yeah. So he intimated there was something not kosher. And that got the investigation going. And then in the second game with him he resigned after one move to kind of further indicate his displeasure. There's been investigation since. There is no conclusive evidence that he cheated.

Steve: Wow.

Leo: But there's a lot of circumstantial evidence.

Steve: Well, and it's also really interesting that when he's in an environment where he cannot receive any help...

Leo: Right.

Steve: ...he's not playing as well.

Leo: So the single thing that they did was they delayed the broadcast. So normally it's streamed live. And so an accomplice at home could be watching the game live and somehow, we don't know how, transmit the move.

Steve: Communicating back to him.

Leo: But if they delay the broadcast by 15 minutes, suddenly his results aren't as good. So it's a little suspicious. It's unknown, really, frankly.

Steve: Yeah.

Leo: And now Magnus has stopped playing in the World Championship. He's said, "I don't want to play anymore." So.

Steve: Those darn computers.

Leo: When computers got that good, we thought that's it for chess. And it's turned out no. There's been some issues; but people, humans still like playing. They even like playing against machines, as good as they are.

Steve: Wow.

Leo: Yeah.

Steve: Okay. So in very welcome news, CISA has announced that they have started scanning the Internet-exposed networks of the U.S.'s critical infrastructure for vulnerabilities and warning those who are responsible. Yay. As we know, we've been covering other countries' welcome announcements of their intentions and results from doing the same, and in some instances their scans have turned up many important things that did need fixing. So it's very welcome news that now, in the U.S., CISA has begun doing the same thing here. CISA's announcement last week is titled "CISA Establishes Ransomware Vulnerability Warning Pilot Program"; and it, too, has already borne fruit.

They said: "Recognizing the persistent threat posed by ransomware attacks to organizations of all sizes, the Cybersecurity and Infrastructure Security Agency announces today the establishment of the Ransomware Vulnerability Warning Pilot (RVWP) as authorized by the Cyber Incident Reporting for Critical Infrastructure Act" - boy, they love their acronyms - "the CIRCIA of 2022. Through the RVWP, CISA will determine vulnerabilities commonly associated with known ransomware exploitation and warn critical infrastructure entities with those vulnerabilities, enabling mitigation before a ransomware incident occurs." Perfect.

They said: "The RVWP will identify organizations with Internet-accessible vulnerabilities commonly associated with known ransomware actors by using existing services, data sources, technologies, and authorities, including our free Cyber Hygiene Vulnerability Scanning service." What? "Organizations interested in enrolling can email vulnerability@cisa.dhs.gov." Then they finished: "CISA recently initiated the RVWP by notifying 93 organizations identified as running instances of Microsoft Exchange Service with a vulnerability called 'ProxyNotShell,' which has been widely exploited by ransomware actors. This initial round of notifications demonstrated the effectiveness of this model in enabling timely risk reduction as we further scale the RVWP to additional vulnerabilities and organizations."

So that's all really good news. And of course this begins as just, oh, we're just sticking our toe in the water to notify people of ransomware because of course who could object to that? We can foresee that this is going to become much broader as it proves itself over time. And I just think this is going to have to become the way things go. Note that the Cyber Hygiene Vulnerability Scanning service they refer to is not open to the private sector unless the organization qualifies as a critical infrastructure provider. In an FAQ, CISA answers the question "Who can receive services?" by replying: "Federal, state, local, tribal, and territorial governments, as well as public and private sector critical infrastructure organizations."

And of course as the guy who created and launched GRC's ShieldsUP! Service 24 years ago, back in October of 1999 - and that was, by the way, 106,477,630 network scans ago - I've seen firsthand how important and effective this sort of proactive scanning can be. And even more recently, Leo, following our Podcast 389, which was January 30th of 2013, I quickly added that Universal Plug and Play, you know, UPnP scanner to ShieldsUP!. And since then it has informed 55,301 visitors that they have, for some reason, Universal Plug and Play publicly exposed.

Leo: Yeah.

Steve: So I think it's very clear that this sort of proactive scanning is where we have to go. It's just going to be so important. And, you know, it's probably the local governments or the regional and national governments that need to do the scanning because their packets need to be above reproach; right? You just don't want anybody scanning organizations because they may be looking for vulnerabilities to exploit as opposed to vulnerabilities to notify the responsible parties. So again, this is a change that has been coming for a while. And, you know, yay.

Leo: Yay, yay, yay.

Steve: Okay. We're going to talk about Flying Trojan Horses after you tell our listeners why we're here.

Leo: ExpressVPN, yeah.

Steve: That's right.

Leo: And we're still working on our Midjourney props. We're trying to get some Flying Trojan Horses. For some reason the AI just refuses to let Trojan Horses fly. They just - they're ground-bound. I don't know why that is.

Steve: This is sort of reality-check time, but I think everyone's going to find this interesting. A large and significant group of fully bipartisan, not just token bipartisan, senators have all co-signed a letter to CISA's director, Jen Easterly. The letter requests that CISA examine the very popular drones made by DJI for evidence that China might be covertly acquiring valuable information from them. Okay. In a minute, we're going to walk through a complete, interesting, and revealing well-conducted technical forensic analysis of DJI's drone controller software to learn exactly what's going on. But let's first set the stage, because this just happened, by looking at this letter which reveals the politics which are driving the concern.

And for those who don't follow politics, the names of the senators won't mean much. But for those who do, these are all senators, many of them senior, that you'll have heard of. So this letter was signed by Mark Warner, Marsha Blackburn, Richard Blumenthal, John Thune, Jeanne Shaheen, Rick Scott, Kyrsten Sinema, Todd Young, J.D. Vance, Ted Budd, Dan Sullivan, Deb Fischer, Mike Braun, Cynthia Lummis, Tommy Tuberville, and Jerry Moran. So serious players here on the Senate.

So here's what the senators are asking of CISA's director. They said: "Dear Director Easterly: We write today regarding the cybersecurity risks posed by the widespread use of drones manufactured by Shenzhen DJI Innovation Technology Co., Ltd. (DJI) to operators of critical infrastructure and state and local law enforcement in the United States. In short, we believe that, given the company's identified connections to the Chinese Communist Party, the use of its drones in such sensitive contexts may present an unacceptable security vulnerability. We ask that the Cybersecurity and Infrastructure Security Agency (CISA) evaluate this concern and make the results of its evaluation available to the public through the National Cyber Awareness System.

"China's efforts to modernize the capabilities of the People's Liberation Army, including through their Military-Civil Fusion strategy, which systematically blurs the lines between PLA and civilian science and technology research and development efforts, are well

documented. In October 2022, the Department of Defense identified DJI as a 'Chinese military company' operating in the U.S. under Section 1260H of the William M. ('Mac') Thornberry National Defense Authorization Act for Fiscal Year 2021. Identification of this relationship between DJI and the PLA suggests a range of risks to U.S. operators of the technology, including that sensitive information or data could wind up in PLA hands. Indeed, Huawei, another entity identified under Section 1260H, has been credibly accused by the Department of Justice of misappropriating intellectual property and trade secret information from U.S. companies. Yet, despite these risks, the use of DJI drones remains widespread throughout the U.S."

Leo: Here we go again.

Steve: Uh-huh. "In 2021 it was reported that DJI controlled almost 90% of the consumer market in North America and over 70% of the industrial market."

Leo: Yeah. It's the only kind I buy.

Steve: I know. They're the best; right, right. And Leo, you're doing your part to increase those percentages.

Leo: Yes, I am. Every time I sink a drone.

Steve: "And in 2019 it was reported that 73% of public safety operations are flown by the company's aircraft. As a result, the CCP may have access to a variety of proprietary information. For example..."

Leo: They don't have my information. For crying out loud.

Steve: Right. Yours is drowned.

Leo: Yeah.

Steve: "For example, a 2017 Department of Homeland Security assessment warned that Chinese companies had used grape production information gathered by a DJI drone purchased by a California wine producer to inform their own land purchasing decisions. Even worse" - oh, worse than that - "the widespread use of DJI drones to inspect critical infrastructure allows the CCP to develop a richly detailed, regularly updated picture of our nation's pipelines, railways, power generation facilities, and waterways." Which I guess they can't get from their spy satellites orbiting overhead, or their balloons. Anyway, they said: "This sensitive information on the layout, operation, and maintenance of U.S. critical infrastructure could better enable targeting efforts in the event of conflict.

"We appreciate that CISA has addressed this risk in the past, most notably in a 2019 Industry Alert stating the federal government's 'strong concerns' with Chinese drones and warning entities to be 'cautious' in purchasing them. However, over the past four years more information regarding the scope of the problem has become available" - and that's what we'll be talking about - "including the official identification of DJI as a Chinese

military company by the Department of Defense. We therefore ask that CISA revisit its analysis of the security risks posed by the use of DJI-manufactured drones and release the results of that analysis publicly through the National Cyber Awareness System."

What do we know about DJI's observed behavior? Three years ago, the security firm GRIMM went to a great deal of trouble reverse-engineering DJI's software. Here's what they found. They said: "Given the recent controversy over DJI drones, a defense and public safety technology vendor sought to investigate the privacy implications of DJI drones within the Android DJI GO 4 application. To conduct their analysis, the vendor partnered with Synacktiv" - and that's a group we've referred to before, a credible security firm - "who performed an in-depth dynamic and static analysis of the application.

"Their analysis discovered four main causes of concern within the DJI GO 4 application, most notably the application contains a self-update feature that bypasses the Google Play Store. The application contains the ability to download and install arbitrary applications, with user approval, via the Weibo SDK. During this process, the Weibo SDK also collects the user's private information and transmits it to Weibo. Prior to version 4.3.36, the application contained the Mob SDK, which collects the user's private information and transmits it to MobTech, a Chinese analytics company. And finally, the application restarts itself when closed via the Android swipe close gesture. Thus users may be tricked into thinking the application is closed, but it could be running in the background while sending telemetry requests.

"To provide an independent review of the findings, the vendor then asked GRIMM" - these guys - "to validate Synacktiv's findings. This blog describes," they wrote, "GRIMM's setup and workflow for validating the Synacktiv research. Using the techniques described in the following sections to perform static and dynamic analysis on the DJI GO 4 Android application, GRIMM was able to verify and confirm the findings from Synacktiv's report. The code associated with this blog post can be found in our GitHub repository."

Okay. So let's follow along because it's much more interesting than just being asked to accept the conclusions without knowing where they came from. It's also interesting to learn how such an investigation is conducted. So they wrote: "GRIMM's researchers used two different set-ups: an ARM-based Android 6.0 Marshmallow (API 23) emulator; and another with two physical devices, a rooted Nexus 6 and an unrooted Motorola Moto 3G.

"The Android emulator is a part of Android Virtual Devices Manager, a subsystem of Android Studio that can be controlled through ADB, which is the Android Debug Bridge. Additionally, Android Studio is able to redirect all traffic to an HTTP Proxy. We redirected traffic through Burp Suite, under which requests can be captured and intercepted. Frida, a dynamic instrumentation tool, was also used on the emulator by directly pushing and running Frida server on the device. We chose API 23 due to the added Certificate Authority certificate protections which were introduced in Android 24." So the point was that their TLS proxying was easier under API 23.

"The Nexus 6P running Android N, API 23 also, was connected to a desktop through USB and controlled through ADB. Both devices were connected to the same wireless network. The setup for analysis on this phone was similar to the emulator, except for the proxy and certificate. The proxy was done with iptables to redirect all traffic on ports 443 and 80 to Burp. Originally, we attempted to connect via USB Ethernet adapter, but we found that the behavior of the app was different from the more normal WiFi setup. We used Frida to bypass SSL pinning on the Nexus 6P. Additional testing was conducted with a similar setup using a Motorola Moto 3G running Android L and the OWASP ZAP proxy."

Okay. So that setup gives them a testing platform, the ability to view, extract, and debug the Android code through the Android Debug Bridge. And they have an effective shim

which allows them to transparently monitor all communications in the clear without encryption so they can see everything that's going on.

They said: "The DJI GO 4 Android application was heavily obfuscated, utilizing both static and dynamic obfuscation techniques to thwart analysis. Synacktiv provided GRIMM with a detailed write-up and scripts to deobfuscate the code and help analyze the application. The first protection the application uses is a custom version of (B-A-N-G-C-L-E) Bangcle. This tool encrypts Java bytecode (.dex files), which can then be decrypted and loaded dynamically during runtime. To understand and defeat this technique, we can draw parallels to the well-known binary obfuscation technique 'packing,' where the code contained within an executable is also decrypted and loaded during runtime.

"The two main methods of deobfuscating packed binaries are to statically analyze the packing routines and extract the data, or dump the memory of the executable after the data has been decrypted. In the context of Android applications, we can do the same. There has been previous research on static analysis of Bangcle. However, Synacktiv was unable to apply the previous techniques to the DJI GO 4 application, as it is using a custom version of Bangcle. Rather, GRIMM utilized Synacktiv's Frida scripts to search through the memory of the Android application at runtime and dump the decrypted .dex Java bytecode files. With the dumped Java bytecode files, GRIMM was able to use Java decompilers such as jadx and Procyon to decompile the bytecode and obtain near-accurate Java source code on which we can perform," they wrote, "static analysis.

"In addition to protecting the Android Java bytecode, the Java source code also features various static obfuscation techniques, most notably string obfuscation. Most of the strings used in the Java source code are obfuscated. However, this protection is rather simple to decipher, as described by Synacktiv." And it turned out they were just Base64 encoded after being XOR-scrambled with a hardcoded key.

And they said: "Additionally, the DJI GO 4 application uses obfuscated string getter classes. These classes define an accessor function which takes an index to the desired string. These obfuscated strings can be easily recovered by decompiling the relevant class, adding a main function that dumps the strings, recompiling the code and executing it." And I'll just note that there's nothing at all nefarious about using string indexes. I did exactly the same thing in my design of SQRL. It's a very clean way of adding language-independence to an application. Throughout your code you refer to UI display strings only by index, and then a language pack provides the phrase dictionary which the indexes point to. Anyway, they said: "With the ability to decompile the Java code and decode strings within the Java code, as well as intercept and analyze the application's network requests, we were able to fully reverse engineer the application's execution."

So Synacktiv's report describes the DJI GO 4's custom update mechanism. This update service does not use the Google Play Store and thus is not subject to the review process. As such, there is no guarantee that the application that is downloaded for one user matches that of another user. If DJI's update server is malicious or compromised by an attacker, it could use this mechanism to target individual users with malicious application updates. And this behavior is a violation of Google's Developer Program Policies, which states: "An app distributed via Google Play may not modify, replace, or update itself using any method other than Google Play's update mechanism. Likewise, an app may not download executable code, for example, dex, JAR, or .so files, from a source other than Google Play."

They said: "Using dynamic analysis, GRIMM researchers were able to intercept traffic pertaining to the update of the DJI GO 4 application. Upon application startup or when using the 'Check for Updates' option within the application, a request was sent to service-adhoc.dji.com, which responds with a URL to an updated application APK. This APK file is

downloaded directly from DJI's servers via a URL." And they provide it. There's a djicdn.com and then a bunch of hex.

"This update option completely bypasses the Google Play Store, giving DJI's servers the ability to fully control the APK downloaded, whether with malicious intent or not. When the server's response is received, the application prompts the user with the update notification. Once the user clicks on the update notification, they're asked to install the update. This update process does require the user to give the DJI GO 4 application the 'Install unknown apps' permission. To help investigate the issue further, GRIMM modified the server's response."

Anyway, I go on in additional detail, as do they, following all of this. And this all comes down to the app, yes, is able to pull updates from anywhere it wants to outside of the Google Play Store. There are some SDKs which do obtain significant personal information which it's easy to argue they don't need. For example, at one point the app grabs the IMEI, the ICCID, the MAC address, the Android ID, the device name and so forth, encrypts it using an RSA public key embedded in this Weibo SDK, and sends it. They basically fully reverse engineered this application. And yes, if you wanted to be worried, then here's an example. What isn't said, of course, is that what other apps are doing the same thing.

They wrap this up finally under "Impact." And they said: "Given these findings, it's useful to consider the best- and worst-case scenarios for how these features are used. While they could just be slightly odd implementations for acceptable behavior, they could also be used in a much more nefarious way." Again, okay.

"In the best-case scenario, these features are only used to install legitimate versions of applications that may be of interest to the user, such as suggesting additional DJI or Weibo applications. In this case, the much more common technique is to display the additional application in the Google Play Store app by linking to it from within the application. Then, if the user chooses to, they can install the application directly from the Google Play Store. Similarly, the self-updating components may only be used to provide users with the most up-to-date version of the application. However, this can also be more easily accomplished through the Google Play Store.

"In the worst case, these features can be used to target specific users with malicious updates or applications that could be used to exploit the user's phone. Given the amount of user's information retrieved from the device, DJI or Weibo would easily be able to identify specific targets of interest. The next step in exploiting these targets would be to suggest a new application via the Weibo SDK or update the DJI application with a customized version built specifically to exploit their device. Once their device has been exploited, it could be used to gather additional information from the phone, track the user via the phone's various sensors, or be used as a springboard to attack other devices on the phone's WiFi network. This targeting system would allow an attacker to be much stealthier with their exploitation, rather than much noisier techniques such as exploiting all devices visiting a website.

"Regardless of whether DJI or Weibo utilize their applications' functionality to target users, they have created an effective targeting system. As such, attackers who know of this functionality may attempt to compromise DJI's and Weibo's servers to exploit this functionality themselves. Given this risk, it's much safer to rely on Google to provide application validation and distribution security." And, okay, I got a chuckle out of that line, since the overtly policy-violating behavior these guys have reverse engineered, demonstrated, and observed for themselves - oh, and I skipped over where they actually changed the URL that was being queried and demonstrated the successful installation of an arbitrary APK. So it can indeed...

Leo: There you go. You can do it. Proof of concept.

Steve: ...install anything, exactly.

Leo: Yeah.

Steve: So anyway, what I got a chuckle out of was that this policy-violating behavior was downloaded initially through the Google Play Store. So, lot of good that did; right? I mean, as we know, there's, what is it, I don't know how - I have it in my show notes a little bit later, how many millions of apps are on Google Play.

So the GRIMM guys conclude, saying this, They say: "This blog post details GRIMM's efforts to validate Synacktiv's privacy assessment of the DJI GO 4 Android application and determine the impact of their findings. After dumping the encrypted classes and setting up an emulated and physical test environment, GRIMM performed static and dynamic analysis in order to reverse the application and validate Synacktiv's findings. The DJI GO 4 application contains several suspicious features, as well as a number of anti-analysis techniques, not found in other applications using the same SDKs. Overall, these features are worrisome and may allow DJI or Weibo to access the users' private information or target them for further exploitation."

Okay. What wasn't directly addressed here was the application platform. This was not the analysis of some controlling code buried in a lawnmower. It's the code controlling what happens to video imaging being captured by the world's most popular aerial drones which are in use, not only by U.S. citizens, but by U.S. law enforcement and military, on military bases in the U.S. and elsewhere. Is there any reason whatsoever to think that anything nefarious is going on? No. Is there any solid evidence of misuse of this technology, perhaps beyond suspicions about some data leakage from grape harvesting? Apparently not. Are the U.S. senators wrong to be concerned? No.

We now have incontrovertible proof that we have unwittingly invited hundreds of thousands of camera-equipped Flying Trojan Horses into our midst, including into areas where there is danger of some of our nation's most private and sensitive operations being sent to a country with whom we appear to be becoming increasingly adversarial.

So this brings us back once again, this time armed with a beautifully clear example, to the utter insanity of the situation we are currently in. None of this makes any objective rational sense if we're doing anything more than merely paying lip service to security. We've walked into it with our eyes wide open. Why? Probably mostly because it was the path of least resistance which was established while everyone was happily getting along and minding their own business. This concrete and clear example begs the question, "But what if?" But this DJI drone instance is just one among millions of similar potential true points of vulnerability.

Okay. For example, the Windows operating system that most of the world is sitting in front of is composed of a kernel and libraries for which Microsoft is the author and has the source code. But in order for it to do anything useful at all, the system also contains countless proprietary third-party device drivers, the source code for which Microsoft has never seen. What do those drivers, in detail, do? No one other than their authors has any idea.

Could any one or more of them have undisclosed nefarious Trojan-like functionality? Of course they could. Why not? And if they don't today, any future update to them could, just like any app in Google's Play Store. For the sake of convenience, an increasing

number of these are included with the base operating system image. But the system also has the capability of going out to fetch additional drivers when needed, and their updates. And all of these many chunks of unknown and unvetted code operate at ring 0 with full unrestricted kernel privileges.

My point is, the actual security model of the world's most pervasive operating system is utterly broken. It's a complete joke. It's smoke and mirrors. We don't want that to be true. It's quite uncomfortable for it to be true. But pretending that it isn't true doesn't change the reality. It's the Wizard of Oz, where we're supposed to keep our eyes on the impressive display of security in front of us - Steve Ballmer jumping around onstage - while we dare not consider and look behind the flimsy curtain where reality lurks.

What's the solution? Well, if ever hostilities across the world escalate into a true fight, we're screwed. The first thing you should do probably is turn off your router and preserve the operation of your own internal network.

Leo: That's a good point.

Steve: The only solution I can see is for everyone to soberly appreciate the true consequences of what would happen now, in today's deeply interconnected world, if superpower hostilities were ever to boil over. We all need to just get along, and for all of the embedded Trojan code that everybody has probably installed over time in everybody else's worlds to remain untriggered and unused.

Looping back to the DJI GO 4 app, that app is just one from among the - and here it is - the 2.65 million apps which are currently listed and available for download through Google Play Store. The economics of what Google has built does not allow for any authoritative representation of app security to be made. Google depends upon some of its own engineering and the engineering of many other security companies to analyze apps and catch misbehavior. But we are constantly learning of hundreds of thousands, if not millions, of downloads of apps by users, which are later found to contain malicious functions.

The only long-term solution, if we're really willing to foot the cost of true security, is for all proprietary closed solutions to be eliminated and for everything to be open source, created by a broad community of cross-checking developers. Until and unless that happens, all we can do is hope for the best.

Leo: Wow, Steve. You've finally come around. I am a big proponent of open, and that's one of the many reasons why.

Steve: Yeah.

Leo: Richard Stallman calls proprietary software "malware" because you never can know exactly what it's doing or who it's phoning.

Steve: No. And as I've said, the idea that voting machines are...

Leo: That's nuts. Closed source.

Steve: Like Dominion? It ought to be open source.

Leo: Open source it.

Steve: You can still make money by selling the hardware.

Leo: Hell, yeah.

Steve: The touchscreens and the machines and all that stuff. You just can't keep what it does to yourself. That is nuts.

Leo: Totally agree.

Steve: And so we've evolved a proprietary software ecosystem around powerful companies that want to leverage the fact that their stuff is secret. And we're now looking, we're in danger of having what we have sown being reaped against us. And again, think about all of the random stuff we install with device drivers running in ring 0, and no one has ever seen what's inside them.

Leo: Yup.

Steve: We have no idea.

Leo: Yup. Really good point. I couldn't agree more.

Steve: And I don't know how we get there from here. But at least taking a sober look at where "here" is, is important. And yes, you know, flying Trojans from a Chinese military-owned company, that's a little nerve-wracking. And, okay, so the government is going to have to use domestic drones that aren't going to be as good and are going to cost three times as much.

Leo: Right.

Steve: But if that's what you want, then that's what you're going to have to pay for.

Leo: And if we go to war, throw your smartphones out...

Steve: Unplug.

Leo: ...and go get those Nokia candy bar phones.

Steve: Yeah.

Leo: You know, we were talking about this earlier on MacBreak Weekly. Russia's government says "no smartphone for you" because smartphones are inherently spy devices. I mean, just in every respect.

Steve: Yup. They are connected computers, and they're not open. They are closed.

Leo: And nobody knows what they're doing, yeah.

Steve: Exactly.

Leo: Yeah. Very good. Thank you, Steve. You've done it again, my friend.