Embedding AWS Credentials

Description: This week we look at Google's just announced and launched open source software vulnerability rewards program. We ask the question whether TikTok leaked more than two billion of their users' records. We look at Chrome's urgent update to close its sixth zero-day of 2022 and at a worrisome "feature" I think it a bug! in Chrome. A somewhat hidden autorun facility in PyPI's pip tool used for downloading and installing Python packages is being used to run malware. And we examine a recent anti-quantum computing opinion from an Oxford university quantum physicist. Then I have two bits of miscellany, three pieces of listener feedback, a fun SpinRite video discovery, and my discovery of a wonderful and blessedly prolific science fiction author. And after all that, we look at the result of Symantec's recent research into their discovery of more than 1,800 mobile apps which they found to be leaking critical AWS cloud credentials, primarily due to carelessness in the use of today's software supply chain.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-887.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-887-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. Did that TikTok hack really happen? Steve says, yeah, probably not. There's a new Chrome zero-day you definitely will want to patch. An Oxford University physicist says quantum computing is bogus. And then Steve will reveal his brand new favorite science fiction author and a new 20-volume series, plus a look at why people are embedding AWS credentials in their apps. It's a bad idea. All coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 887, recorded Tuesday, September 6th, 2022: Embedded AWS Credentials.

It's time for Security Now!. Get ready to protect yourself. Put on your hard hat. We've got some construction to do with Mr. Steve Gibson of GRC.com.

Steve Gibson: Put your hard hat on your hard head.

Leo: A hard hat with a propeller, if you have it.

Steve: You're listening to this podcast, you're not soft.

Leo: Line it with tinfoil, you'll be set.

Steve: So here we are at the beginning of September, Security Now! Episode 887. And, okay. So I didn't have a topic for most of the setup for this until I ran across a report that Symantec just published about their findings - okay, I should finish that thought - their findings for the cloud security of mobile apps. And I was a little confused because they were talking about supply chain. And I thought, how is mobile app security about supply chain? And now I get it, and it's really interesting. So it became - so I moved things around. I was going to start talking about this grumbly physicist from Oxford who doesn't think that - he's a quantum physicist who doesn't think that this quantum computing stuff is ever going to come to anything.

Leo: Oh, music to my ears. I've been grumbling about this for a long time.

Steve: I know.

Leo: And I'm no physicist, so that's - yeah, yeah.

Steve: No. But Leo, when factoring 33 is an achievement...

Leo: Yeah, exactly. Exactly, yeah.

Steve: Okay. Anyway, so that got moved to the end of our discussion of all the other stuff, and Symantec's story now takes the lead, which puts it at the far end. But we've got a lot of other cool things to talk about first.

So we are going to take a look at Google's just announced and launched latest open source software vulnerability rewards program. We ask the question whether TikTok leaked more than two billion of their users' records. We look at Chrome's urgent update to close its sixth zero-day of 2022, and at a worrisome, uh, feature? I think it's a bug in Chrome. I have a little demo for our listeners which is a little bit unnerving. Everyone can do it. I also have a somewhat, well, news of a somewhat hidden autorun facility in PyPI's pip tool, which is used for downloading and installing Python packages. That feature is being used to run malware, to the surprise of people using pip. And as I said, we're going to examine a recent anti-quantum computing, we'll call it an "opinion" - it borders on a rant - from an Oxford University quantum physicist.

Then I've got two bits of miscellany, three pieces of listener feedback, a fun SpinRite video discovery from this morning, and I'm most frankly excited about this than anything else: my discovery of a wonderful and blessedly prolific sci-fi author. And after all that, we're going to look at, as I said, Symantec's research into their discovery of more than 1,800 mobile apps which they found to be leaking critical AWS cloud credentials, primarily due to the carelessness in the use of today's software supply chain. So I don't think our listeners are going to be bored.

Leo: No. In fact, I've been waiting for this show all week. There's so much to talk about. The funniest thing happened on the radio show. On Sunday, somebody called and said...

Steve: Oh.

Leo: You already know what I'm talking about.

Steve: The false positive?

Leo: Yes.

Steve: Oh, no.

Leo: And I said, what? Yeah, because apparently a lot of Windows users suddenly thought they were, in fact all Windows users suddenly thought they were infected. Anyway, just so many stories, so little time.

Steve: Okay. So this is just so classically xkcd, I love it. It's titled "General Physics Safety Tip." And it's a very small flow chart, very simple flow chart which is designed to help you decide where to stand. It answers the question and poses it, "Should I stand near this thing?"

Leo: It's a simple question, but an important question.

Steve: Oh, my god. Well, yes. And then so it poses that question in the top box, which feeds into the decision triangle, or I'm sorry, the decision diamond, where the way to answer the question "should I stand near this thing" is to ask, well, "Are physicists excited about it?" And if the physicians are not excited about it, then maybe you should stand near it. If physicists are excited about it, then, no. Do not get...

Leo: No. No, definitely not. Get away. Run. Uh-oh. Steve froze. "In general," says Retcon5, "avoid exposure to any temperatures, pressures, particle energies, or states of matter that physicists think are neat." I love it, Steve. That's awesome. Randall is so good. You know, I interviewed him on a Triangulation some years ago.

Steve: I'm so glad you did. And I was thinking, like, this is the kind of stuff that we would never have were it not for the Internet.

Leo: Yes.

Steve: I mean, the Internet brings all kinds of crap.

Leo: God bless the nerds, yes.

Steve: You know, it brings all kinds of crap along with it. That's just inevitable. But like this, before the Internet we had magazines that were monthly.

Leo: Right, right. But they weren't up to date.

Steve: No. And the New Yorker would have some great cartoons that would like stand out. But you didn't have like a - you couldn't get an IV of this stuff. This is just...

Leo: Yeah. Randall Munroe, it was back in 2019, Triangulation 412, if you haven't heard it yet. Great interview with the creator of xkcd. Yeah, I agree. Thank goodness the Internet's given us some real nice things, and that's one of them.

Steve: And you think about it, like, the bounds and the depth of his creativity, like where did this come from? Right? I mean, like he could easily think, well, I don't have any good ideas, and then here's like this, which is just wonderful.

Leo: Right. And in days gone by, where would you go? Would you go to a newspaper and say I'd like to publish this comic? And they go, no, no one will understand this.

Steve: Right.

Leo: This guy with his genius capabilities probably would be underemployed and no one would know about him. So, yeah, yeah, it's great.

Steve: It's why when people have asked me, like, for some career advice, I've said to them, I would specialize. That is, become, invest however many tens of thousands of hours are necessary to become the best something. And it doesn't matter what. Well, I mean, it has to have some application. But the point is...

Leo: Well, not now, with the Internet, not even then. Not even then.

Steve: No, that's true.

Leo: You know? If a thousand other people might be interested, that's enough.

Steve: Yes. And the idea is that, if you were like just the unbelievable best plumber, well, in the old days, well, all of the toilets in your city would be working now.

Leo: You'd own them, yes.

Steve: But you wouldn't have much marketing reach beyond that. Now, you know, the Vatican might hire you because they really need your help.

Leo: Exactly. I think that's brilliant, Steve. Yes.

Steve: And they could find you. That's the point.

Leo: Exactly.

Steve: They would be able to find - they'd just put into Google, who is the best plumber? And up would come Maury Mergensteen. And they'd go find him and say, Maury, we need some help here with the Pope's commode.

Leo: Actually, you just told the story of Father Robert Ballecer. That's, in a nutshell, the secret to his success. So there you go.

Steve: Yeah. Okay. So last week Google announced and launched a new Google-targeted Open Source Software Vulnerability Rewards Program. I say it's Google targeted because it's for their open source software project, but okay. Their money's good. So this is what they said. And yes, it's a little bit commercial, but it's still interesting. And after all, they're giving away money.

So they said: "Today, we are launching Google's Open Source Software Vulnerability Rewards Program" - which, because it's a mouthful, you can shorten as OSS VRP - "to reward discoveries of vulnerabilities in Google's open source projects. As the maintainer of major projects such as Golang, Angular, and Fuchsia, Google is among the largest contributors," they write, "and users of open source in the world. With the addition of Google's OSS VRP to our family of VRPs, researchers can now be rewarded for finding bugs that could potentially impact the entire open source ecosystem.

"Google," they write, "has been committed to supporting security researchers and bug hunters for over a decade. The original VRP program" - or just VRP - "established to compensate and thank those who help make Google's code more secure, was one of the first in the world and is now approaching its 12th anniversary. Over time, our VRP lineup has expanded to include programs focused on Chrome, Android, and other areas. Collectively, these programs have rewarded more than 13,000 submissions, totaling over $38 million paid." That's pretty good, on average.

They said: "The addition of this new program addresses the ever more prevalent reality of rising supply chain compromises." And actually we're going to be winding up talking about supply chain compromises of a different ilk. They said: "Last year saw a 650% year-over-year increase in attacks targeting the open source software supply chain, including headliner incidents like Codecov and of course Log4j vulnerability that showed the destructive potential of a single open source vulnerability. Google's OSS VRP is part of our $10 billion commitment to improving cybersecurity, including securing the supply chain against these types of attacks for both Google's users and open source consumers worldwide."

And finally, "Google's OSS VRP encourages researchers to report vulnerabilities with the greatest real and potential impact on open source software under the Google portfolio. The program focuses on all up-to-date versions of open source software, including repository settings, stored in the public repositories of Google-owned GitHub organizations" - so Google, GoogleAPIs, GoogleCloudPlatform, and so forth. And, and this I thought was surprising and significant, "those projects' third-party dependencies, with prior notification to the affected dependency required before submission to Google's OSS VRP."

So if you find a problem not only in Google's own stuff, but in something they're pulling into their project and are dependent upon, that qualifies, too. So you fix it there, and then you say hey, Google, I found something that just helped you. Pay up. So, and they finish: "The top rewards go to vulnerabilities found in the most sensitive projects." Oh, I did get a kick out of this. "Depending on the severity of the vulnerability and the project's importance, rewards will range from $100 to $31,337." And of course 1337 is hacker-esque, it's LEET upside down. So, yes, a little tip of the hat to the hacker community.

Anyway, so I've got links to where to go to find out more information. It's generally at bughunters.google.com. You can start there and then browse around. They've got rules for qualifications and so forth. But overall, what occurred to me as I was reading through this and choosing to include it to start off today's podcast, is the degree to which bug bounties have become a part of today's modern software ecosystem. It's no longer a surprise for a company to be offering a bug bounty for the discovery and responsible reporting of problems found in their software.

But it wasn't so long ago that this was unheard of, or that it was an enlightened exception. Today it's the way large companies do business. They figure if we can't find all of the important bugs ourselves, and what we're seeing is that apparently no one can, then you reward the white hats who do. And more significantly, as we've seen, this sort of bug hunting, while not guaranteeing a steady paycheck, at least not at the start, could increasingly be considered a valid and workable career of sorts.

So very cool that, I mean, I'm glad to see Google's doing this. They've got the money to do it. They're going to get the benefit. And again, they've obviously got a very capable stable of internal security researchers and bug hunters. Yet even so, they're saying we'll take any help we can get, which is really the mature thing to do in this day and age.

Leo: It's also a defensive move because, as we've talked about many times, there's places like Zerodium offering big bucks for Google flaws, as well.

Steve: Ah, and not fixing the packages, but selling them to, yes...

Leo: To bad guys, yeah.

Steve: To the state-sponsored, yes, bad guys. Yeah. Good point. So did TikTok leak 2.05 billion, with a "b," user records? TikTok says no, but other independent researchers are not so sure. Every week while I'm looking through the past week's news, half, really half of what I see are breach reports. You know, this or that company reports that it was breached, and bad guys may have obtained the data for typically a couple hundred thousand of their users, you know, more or less. And okay, that's not good. But there's generally not much more to say about it. It's a bit like this or that company got hit by ransomware. Not good. We're sorry. Hope you recover. But the potential exposure of more than two billion user records by TikTok, well, we're talking about this one because of who it is, and the size, scale, and impact of the possible breach.

When news of this appeared last week, TikTok pressed their "Respond to the press" button, and out popped a statement reading: "TikTok prioritizes the privacy and security of our users' data. Our security team investigated these claims and found no evidence of a security breach." Then the button popped back out. But TikTok's canned-sounding denial follows reports of a hack that surfaced on the Breach Forums message board Saturday, with the threat actor noting that the server holds 2.05 billion records in a single humongous 790GB database.

The hacking group known as BlueHornet, also known as AgainstTheWest, or ATW, tweeted: "Who would have thought that TikTok would decide to store all their internal backend source code on one Alibaba Cloud instance using a trashy password?" Bob Diachenko, who's known as the open database hunter, and he's a threat intelligence researcher at Security Discovery, he said the breach is "real," in quotes, I mean, like that's exactly what he said, "real," and that the data is likely to have originated from "Hangzhou Julun Network Technology Co., Ltd. rather than TikTok." But Troy Hunt wasn't yet convinced. Troy tweeted: "This is so far pretty inconclusive. Some data matches production info, albeit publicly accessible info. Some data is junk, but it could be non-production or test data. It's a bit of a mixed bag so far."

And then just before I put everything together, I checked both of their Twitter feeds for any updates. Bob Diachenko's feed had two updates. The first said "Update: While there is definitely a breach, it is still work in progress to confirm the origin of data, could be a third party." That was followed up sometime later with "Update 2: Okay, #TikTokBreach is real. Our team analyzed publicly exposed repos to confirm partial users' data leak."

But then Troy retweeted a tweet and added: "The thread on the hacking forum with the samples of alleged TikTok data has been deleted and the user banned for 'lying about data breaches.'" The group that was banned, as explained in the tweet that Troy retweeted, was that same BlueHornet/AgainstTheWest who made the original allegation. So it appears that the whole thing was, as TikTok originally claimed, not a breach at all.

Though this story sort of cancels itself out, I wanted to share this because it's a great example of what is continually going on behind the scenes in the security industry. Not all players are on the up and up. And not everything - surprise - that's tweeted is factual. And Troy, for whom this is certainly not his first rodeo, knows to remain skeptical until all the facts are in and proven. So...

Leo: Yeah. You know, we have to deal with this all the time. A lot of other publications hoping to get the links will go with this. When I saw the story, one of the first people I looked at was Troy Hunt.

Steve: Yup.

Leo: His skepticism immediately stopped me cold. It's very easy for some jerk to claim this, get a lot of attention. And worse, a lot of publications will just jump on it because they know that's how you drive traffic. So as any journalist, you've really got to be - we're very skeptical about all this stuff until it's proven to be the case. And we do not jump on these things.

Steve: Yeah. And in fact even Brian Krebs, you know, who is a well-known great journalism reporter within this whole infosec...

Leo: I'm not going to say "great" anymore. Go ahead. Keep going.

Steve: Okay. Well, what I was going to say...

Leo: Because he got in big trouble with this Ubiquiti thing.

Steve: And that's exactly what I was going to say, was that he did, like, say, you know, I had a single source. I went with it. And I apologize to Ubiquiti and had to retract it.

Leo: Because he got sued, and it's pretty much widely believed that this was part of the settlement with Ubiquiti. And in fact he would have lost because he knew the guy was a hacker and continued to run the story when he should have immediately said, "I was wrong. This guy was the guy who perpetrated the hack. He used me to, basically, to extort Ubiquiti." And in fact, you know, yeah. So I think Brian, I'm sorry, but Brian's got a black mark in my book now. I'm not sure I'm going to fully trust him going forward.

Steve: Yeah, little bit too much inertia behind the story, I guess.

Leo: Again, a case of I think somebody very anxious to get clicks, and as a result not being very thoughtful.

Steve: And unfortunately, you know, if clicks are your model, then...

Leo: You have to.

Steve: ...it's really tough to say, uh, whoops. That's not what happened.

Leo: Yeah. So Krebs has fully retracted now, taken all those stories down. But in the kind of anodyne language that tells me this is part of the settlement, you know.

Steve: I see. So not a huge mea culpa.

Leo: No.

Steve: Just, okay. Well, an urgent Chrome update required and got an urgent patch. Last Friday, Chrome bumped up to version blah blah blah ending in .102 to urgently close a vulnerability that was being actively exploited in the wild. As usual, little more is being said of CVE-2022-3075, which involves, all we know is insufficient data validating in Mojo, which is a library of routines to provide platform-agnostic interprocess communication. Google credited an anonymous researcher with their report of the high-severity flaw on August 30th. And I'm again impressed by the Chromium team's three-day incident response. To learn about it on the 30th and push it out on Friday the 2nd, that's great.

So this is zero-day number six for the year in Chrome. And while it's not likely to be an emergency for everyone, everyone as always is advised to be sure they're running that version ending in .102. The update applies to the desktop versions of Chrome for Windows, macOS, and Linux. And as always, the users of Chromium-based browsers Edge, Brave, Opera, and Vivaldi are also advised to look around for updates for theirs when they're available.

Okay, now, Leo. If you've got a version of Chrome around, you want to go to webplatform.news, W-E-B-P-L-A-T-F-O-R-M dot news. And what comes up is an innocuous-looking page. But it just puts something on your clipboard without your permission. So you now open like Notepad or something and hit CTRL+V to paste. And you will see a message reading: "Hello. This message is in your clipboard because you visited the website Web Platform News in a browser that allows websites to write to the clipboard without the user's permission. Sorry for the inconvenience. For more information about this..."

Leo: Wow.

Steve: Yeah, huh.

Leo: No, it did not happen to me. I'm not using Chrome. Does this happen on all browsers?

Steve: I tried it on Edge; and, yes, it's got to be a Chromium-based browser.

Leo: Ah. Let me try it on Edge, okay. I was using Firefox.

Steve: Not Firefox. Ah, good for you because Firefox - now, Firefox has a related problem, and this wasn't clear to me as I was tracking this down. So again, to all of our listeners, in a Chromium-based browser, Chrome or Edge, Brave, Opera, Vivaldi that I was just talking about, webplatform.news. And then, like, open Notepad and hit CTRL+V to paste. And you'll get a happy little message planted onto your clipboard.

Leo: Oh. Right you are, my friend.

Steve: Yeah. So hopefully to our listening audience it's needless to say this is not safe.

Leo: Yeah.

Steve: And more than being unsafe, some consider it to be a major security issue. The problem is that the browser's interaction with the clipboard is somewhat tricky. And web developers have been tinkering around with it, considering the deliberate addition of some non-interactive access. You know, it seems pretty clear to me that a user should have to clearly highlight and mark something on a web page, then explicitly issue some form of clipboard copy command for their browser to be given permission to modify their system's clipboard.

Now, and what it feels like to me is the web designers must be saying, hey, well, other native first-party OS apps are able to put something on the clipboard if they want to. Why shouldn't a browser? Why should it not be equally entitled? Well, the answer is, you know, browsers are out hitting random pages, pulling ads in from god knows where, running scripts from who knows who. And all of that should have the ability to put stuff on your desktop clipboard without your knowledge or permission? I don't think so.

So, okay. So, you know, I could see the benefits of having a web page announce that something has already been placed on the user's clipboard, you know, if that was like clearly in their benefit; if they really wanted that to happen. But unfortunately it offers bad guys far too much opportunity for carnage. And there appears to be some lack of clarity on this front. Web developer Jeff Johnson said that what he's calling the "clipboard poisoning attack" was accidentally introduced in Chrome version 104. Okay, and I don't know what 104 he's talking about. We just got 102, and that's what I'm running, and it's in there.

But looking over the pertinent Chromium discussion thread actually kind of muddies the water. I have a link in the show notes to the discussion thread. But, for example, last Monday, that is, Monday before last, not yesterday, eight days ago, from Microsoft on August 29th, a Microsoft Edge person using the Chromium Engine says it's pri-3, which I guess means Priority 3, because this behavior has been there since we shipped async clipboard APIs. It is not a regression. However, I agree that this should be fixed. What? Okay. So "not a regression" means it isn't something that we broke. But this guy is saying I agree this should be fixed, and we should send a breaking change email to blink-dev to figure out the right process to add the transient user activation restriction to the APIs.

And then he says: "I guess pri-1 makes sense since Firefox and Safari are also considering adding the transient user activation," and then "(instead of a user gesture requirement)," which is what they have now. The reason, Leo, it didn't happen to you under Firefox is due to the deliberate presence of something they're calling the "user gesture requirement." Meaning you have to do something in order to, like, enable this event.

Leo: Now, it is writing, not reading. I mean, reading the clipboard...

Steve: Correct, correct.

Leo: ...would be very problematic.

Steve: Correct.

Leo: What is the hazard of writing to my clipboard?

Steve: Okay. So...

Leo: I can see the usefulness of it.

Steve: Yes.

Leo: But I don't know what the problem is.

Steve: So Jeff says, for example, while the problem exists in Apple, Safari, and Mozilla Firefox, as well, what makes the issue more severe in Chrome, this is a requirement for a

user gesture to copy content to the clipboard is currently broken. Jeff appears to be - okay. So I had this here somewhere. He says, oh, the idea that the danger of this is not glaringly apparent to the web developers is a bit surprising. So, okay, I sort of got thrown off here.

Leo: I apologize. I shouldn't interject. I'm sorry.

Steve: Because I did, yeah, I have that covered here.

Leo: Okay. Go back to your - you can answer - put a pin in it and get to that.

Steve: So we're going to get to your question.

Leo: Thank you.

Steve: So last Tuesday on August 30th the Chromium guy in this thread said, "To be clear, READING" - he had it in all caps - "from the clipboard always requires a permission," he says, "very much like geolocation, microphone, et cetera." And he said, "To see what this looks like, check out this demo site." And there's a link to async-clipboard-api.glitch.me. Then he says: "Writing plain text or images to the clipboard can currently be accomplished without a permission or user gesture, although the site and tab in question must be foregrounded." He says: "We are looking to tighten up the security model here."

And he says: "Neither one of these behaviors has changed recently, nor does the new tab page test rely on permission-less, gesture-less clipboard access." Anyway, so at this point the web developer who thinks this is an issue, Jeff Johnson, says that Safari and Firefox are considering making this smoother, not requiring there to be a gesture. So Jeff appears to be saying that Safari and Firefox require the user to have some interaction with a page, though not necessarily a clipboard copy.

He does say that clicking a link or pressing the arrow key to scroll down gives the website permission to overwrite your system clipboard. So you don't even need to do - you just need to have any interaction with the page in order for that permission currently in Safari and Firefox to be given. And they're considering synchronizing themselves with Chromium where even that's not necessary. So although the Chromium guys are saying, uh, and even the Microsoft guy is saying, maybe we'd better rethink this.

So the idea that the danger of this is - okay. So blah blah blah blah blah. Okay. So aside from being annoying and worrying, we have the ability of any web page to replace my system's clipboard data without my permission. Okay. If I were using my computer, and I had something on the clipboard, and then I went to paste it somewhere else, and I got some message, I got some, like, image or text that I had never seen before, I would be sure that my machine had been infected with malware of some kind that had messed with my clipboard without my permission. Because I would be freaked out.

Leo: That's somebody who doesn't understand how the clipboard works because it happens all the time. Password managers wipe the clipboard so your clipboard can't be read with the password on it. It's not at all unusual for clipboards to have multiple different versions of the content, depending on where you're pasting, and do

content-aware paste. So you might get an image in some case, you might get a text in another case. The clipboard is often manipulated by the OS. This is not at all unusual.

Steve: So you're saying if you pasted the contents of your clipboard, and it was something you had never seen before, and you had never pasted into the clipboard, that wouldn't concern you?

Leo: Well, yeah, I mean...

Steve: Come on.

Leo: I guess it shouldn't be something random. But clipboards are often manipulated by the operating system. That's not unusual.

Steve: By you. By user hitting edit copy.

Leo: No, no, no, no, no. No, no, no, no, no. I mean, LastPass wipes the - how many times do you have a password on your clipboard that gets wiped after 10 seconds, or you set the time. It gets wiped automatically. There's things happen to your clipboard all the time. And as I said, clipboards have, at least on a Mac, probably not on Windows...

Steve: Okay. So you're saying you would have no problem if random pages that you visit are writing to your desktop clipboard.

Leo: Well, usually they do that for utility; right? So they'll paste something in there that you're going to want - a link or something that you're going to want to paste somewhere else. So they'll try to do it for utility. That's why this feature exists. All right, so it's scaring you. I got it. Is it hazardous?

Steve: Certainly. You could imagine that a site or page manipulates you so that you have a cryptocurrency address on your clipboard, which the web page changes behind your back without you knowing it.

Leo: That's a good - there's a good malicious use. But we say all the time you shouldn't trust your clipboard; right? I mean, we say that all the time.

Steve: Okay. So the issue is no user permission. I mean, no user action at all. You go to The New York Times, and an ad on The New York Times...

Leo: Is put on your clipboard, yeah, yeah.

Steve: No, puts anything it wants.

Leo: Right.

Steve: Not the ad. It puts anything it wants in your clipboard. It just, to me, I mean, the good news is, now that this has come to light, the web designers are saying, oh, okay, we need to do something. I'm sure that they thought it was cool that you'd be interacting with a web app, and the web app would say, okay, we're all finished doing what you wanted. The results are waiting for you on your clipboard. I don't have any problem in using Google Docs and marking something and then hitting CTRL+V, and it gets pasted to my clipboard. That's what I want.

But I have a problem if I go to some website, and without knowing - because I'm like you, Leo. I mean, I'm using my computer. I know what's on my clipboard. And typically I'm copying and pasting things. And yes, I've often had the experience, sometimes it's annoying, in fact, that Last Pass has erased a password before I was able to paste it somewhere.

Leo: It's always annoying. Right. But it's for good reason.

Steve: Yeah. And so I go, okay, yeah, fine.

Leo: Yeah, no, I think, okay, yeah, I can see the hazard here. That was a perfect example of, you know, because you can't really - a crypto account number is so long, you might not remember the one you had cut, and instead put a different one in. That would be a big problem.

Steve: And normally you don't even attempt, it's like a long password, you don't even attempt to memorize it.

Leo: To type it, yeah.

Steve: And then make sure that it...

Leo: So we use copy-and-paste for something like that all the time, yeah, that's a good point, yeah.

Steve: Right. So anyway, the good news is that these guys are looking at it. I just thought it was a little jarring to just go to a page and have it change our clipboard without us giving permission. And so to me it's - just because, I mean, yes, it's true that any app that we have on our desktop could do this. But we would consider it a misbehaving app if it was changing our clipboard in a way that didn't benefit us. And we know what's on the Internet, you know. I would say way less than half of it benefits us. So I don't want, you know, pages that I happen to encounter or components of pages like an ad to run some script that changes my clipboard. That's just like, hands off my clipboard.

Leo: I think there's utility, and that's why they put it in.

Steve: Yes.

Leo: But I can see that potential hazard, yeah.

Steve: And, yeah, they're wanting it to be a first-class citizen of the desktop. And my only reservation is, not everything that lands in my browser is something that I want to trust my desktop with.

Leo: You actually raised probably the real reason why Google did this: Google Docs. Which is a browser-only experience, but they want it to be like a desktop app; right? Or Gmail.

Steve: Yup, yup, yup. And I use the crap out of the Docs cross-clipboard ability. In fact, oh, one of my biggest peeves is that you cannot copy an image out of Google Docs and move it somewhere else. It, like, refuses to let you have it for some reason. It just drives me nuts. But you have to jump through hoops to do that. But anyway.

Speaking of unwanted features - oh, actually, speaking of my throat being dry.

Leo: I've got a wanted feature. Another fine commercial.

Steve: We weren't speaking of that because I can't speak. PyPI is the Python language's popular Package Index repository, inviting alliteration. Python Package Index, thus PyPI. And in another discovery that would expose developers to increased risk of a supply chain attack, it was found that nearly one-third of the packages in PyPI trigger an automatic code execution upon downloading them. Now, sort of as with auto-copying to the user's clipboard, if it's what the user wants, then fine. Having a Python script autorun after downloading a well-designed and benign package, well, that's just convenience; right? It just, like, if you say "pip install," you want it to do that.

But a researcher at Checkmarx noted in a technical report they published last week that a worrying feature in pip's command allows code to automatically run when developers are merely downloading, not necessarily installing, a package. He added that the feature's alarming because, he said, "a great deal of malicious packages we are finding in the wild use this feature of code execution upon installation, or download, to achieve higher infection rates." Yeah, that would follow.

Anyway, one of the ways by which packages can be installed for Python is by executing the "pip install" command, which in turn invokes a file called "setup.py" which comes bundled along with the module. "Setup.py," as its name implies, is a setup script that's used to specify metadata associated with the package, including its dependencies. It provides some of the welcome automation that makes package management convenient in this environment. And in what amounts to a documentation flaw, meaning that it was undocumented and actually surprising, a cautious user might opt to use the safer appearing pip download command, since its documentation states: "Pip download does the same resolution and downloading as pip install; but instead of installing the dependencies, it collects the downloaded distributions into the directory provided, which defaults to the current directory."

In other words, the command can be used to download a Python package without having it installed on the system and all of its dependencies. But as it turns out, executing the download command also runs the embedded "setup.py" script, resulting in the execution of whatever malicious code it might contain. It turns out that "setup.py" autorunning only occurs when the package contains a tar.gz file instead of a wheel file, which is .whl. Although pip defaults to using wheels instead of tar.gz files, attackers take advantage of this behavior to intentionally publish Python packages, you know, malicious Python packages, obviously, without a .whl file, leading to the execution of the malicious code present in the setup script. The Checkmarx report noted that: "When a user downloads a Python package from PyPI, pip will preferentially use the .whl file, but will fall back to the tar.gz file if the .whl file is not present."

At the moment, there's not much that Python users can do. If pip is used to either install or download a PyPI package, bad guys can arrange to run their "script.py" on the user's machine. And given all the recent troubles with supply chain attacks, that's a bit nerve-wracking. So anyway, I just wanted to bring that to our listeners' attention. Python is popular and only becoming more so for many good reasons. And so this looks like a problem that needs to get addressed somehow.

Leo: Yeah. This is also a problem with some Linux distros. Arch, for instance, has a user repository that if you - you don't have to, but many people use an Arch User Repository installer that will run the script. And the better ones say, no, no, you've got to look at the script before you run it. And there are plenty of sites you go to, and you install, you know Homebrew is one for the Mac and Linux. It's a package manager. And you install it with a curl command. You know?

Steve: Right. Ohhh.

Leo: And, you know, they even say, you know, we know this is a terrible way to do it. But, you know.

Steve: But here's the command.

Leo: Here's the command.

Steve: It's easy.

Leo: It's a lot easier.

Steve: Oh, goodness. Yeah. We had some fun years ago talking about dangers of curl. Whoa. Well, and there have been some vulnerabilities in it, too. Okay. So we've been talking about quantum computing recently. Even though the capability to break our current public key cryptographic security protocols remains purely theoretical, and I mean entirely theoretical, the existence of technology that could do so could render the asymmetric cryptography which we depend upon to manage our symmetric keys useless for that purpose.

In practice, the security of anything and everything we currently protect with certificates, and the public handshakes we make to negotiate secret keys, would be open to

circumvention and abuse. Astonishing to me as it is, although bombs are not flying through the air between hostile superpowers, there is clearly very active continuous and slowly escalating cyberwarfare being conducted among and between the world's hostile superpowers. The only thing keeping this under control and at parity is that none of these superpowers has meaningful superiority in cyberspace or, as Leo would never say...

Leo: Don't do it. Don't do it. Don't do it.

Steve: In cyber.

Leo: <yelping>

Steve: If quantum computing's promise is realized, someone will have it first. And that someone will have a massive destabilizing power over the entire rest of the world. The degree to which we depend upon the stabilizing force of today's status quo should not be overstated. I don't think you can overstate it.

And it's for this reason that researchers in cryptography, armed with an understanding of what a future working quantum computer might be able to do, they've already been at work, as we know, we've talked about it recently, for several years on the design and implementation of next-generation so-called "post-quantum" replacement cryptography.

The website "Futurism" runs a column called "The Byte," and last Saturday's title was: "Oxford physicist unloads on quantum computing industry, says it's basically a hype bubble." Now, it would be a full-time job to know everything about what's going on in quantum computing. And I suppose that's this guy's job. And that's good because I'm already overbooked. At the same time, I don't know anything about what biases this guy might have. He certainly seems disgruntled. You know, maybe he applied for some grant and didn't get it. I don't know. But I found the synopsis of what he wrote to be interesting and worth sharing.

So this was in "The Byte" column on futurism. They wrote: "Oxford quantum physicist Nikita Gourianov tore into the quantum computing industry this week, comparing the 'fanfare'" - they have in quotes because that was his word - "around the tech to be a financial bubble in a searing commentary piece for the Financial Times." Now, Financial Times is behind a high paywall or I'd have gone to the source. But I tried, and I couldn't get there. They said: "In other words, he wrote, it's far more hype than substance. It's a scathing, but also perhaps insightful, analysis of a burgeoning field that, at the very least, still has a lot to prove. Despite billions of dollars being poured into quantum computing, Gourianov argues, the industry has yet to develop a single product that's actually capable of solving practical problems."

And, now, he doesn't even mean crypto. Crypto turns out to be an extremely high bar because you can't have any fuzziness. And fuzziness seems to be a side effect of quantum, at least now. Anyway, he says: "That means these firms are collecting orders of magnitude more in financing than they're able to earn in actual revenue," he says, "a growing bubble that could eventually burst." Gourianov wrote for the Financial Times: "The little revenue they generate mostly comes from consulting missions aimed at teaching other companies about 'how quantum computers will help their business,' as opposed to genuinely harnessing any advantages that quantum computers have over classical computers."

Okay, now, for my part, while I think this is an interesting opinion from someone whom others apparently believe knows something about quantum physics, I'll just note as a counterpoint that something is impossible, right up until the time it isn't. And this doesn't guarantee that that something will ever not be impossible. But when the stakes are as high as they are, this sort of tax-deductible research is easy to justify to a company's board of directors.

Anyway, Nikita Gourianov went on to say: "Contemporary quantum computers are 'so error-prone that any information one tries to process with them will almost instantly degenerate into noise,' which scientists have been trying to overcome for years." He also took aim at other assumptions about the field, arguing that fears over quantum computers being able to crack even the most secure cryptographic schemes are overblown. And notably, Gourianov's rant in the Financial Times comes just weeks after a group of researchers found that a conventional computer was able to rival Google's Sycamore quantum computer, undermined Google's claim in 2019 of having achieved "quantum supremacy."

Recalling the sentiment "There's gold in them thar hills," despite the industry's lackluster results to date, investors are still funneling untold sums into quantum computing ventures. You know, yeah, because what if?

Gourianov said: "In essence, the quantum computing industry has yet to demonstrate any practical utility, despite the fanfare." He says: "Why then is so much money flowing in? Well, it's mainly due to the fanfare." The money, he argues, is coming from investors who typically don't have "any understanding of quantum physics," while "taking senior positions in companies and focusing solely on generating fanfare." So in short, Gourianov believes it's only a matter of time until the quantum bubble will pop, and then the funding will dry up.

So anyway, I wanted to share this presumably informed perspective, since many tech media outlets covered this rant in the Financial Times - the Financial Times is well regarded - because this Oxford University quantum physicist may know what he's talking about, because on some level it does appear to fit the evidence, after all, and because it serves as an interesting counterpoint to what does indeed, despite huge expenditures, still seem to be quite a long ways off, if it is even ever practical. So will this quantum crypto panic ultimately turn out to have been misplaced? Maybe. But the stakes are clearly so high that a great deal of wealth is being transferred.

And Leo, as I said at the top, if factoring the number 33 is considered a huge achievement just recently, and if as Gourianov appears to believe this particular branch of quantum physics is not about to be visited by a breakthrough, then the crypto industry probably has time to get its post-quantum crypto right the first time. And I think that's good. Even if "quantum" never happens, the replacement of our aging quantum-unsafe crypto only makes sense. You know? Why not do it?

And since the replacement of everything we have now will take significant time, I'm very glad we're already working to determine what that replacement will be. And it'll be interesting to see whether it is the replacement of pre-quantum crypto with post-quantum crypto that finally bursts the quantum hype bubble. Maybe if like no one suddenly any longer has any cause to worry about achieving a crack in current crypto, then it's like, oh, well, okay, isn't weather forecasting already good enough?

Leo: I mean, yeah, it doesn't hurt to come up with better crypto techniques. I'm using...

Steve: No, exactly. And we know how long it's going to take. It's going to take forever.

Leo: Yeah. I'm using ECC and, you know, some other - like I use that, what is it, that 25519 now for my SSH keys.

Steve: Yup, yup.

Leo: Doesn't hurt; right?

Steve: Well, those are all crackable.

Leo: Oh, right. That's right.

Steve: Those are all, I mean, anything public, anything public key.

Leo: Today.

Steve: So elliptic curve is public key. 25519 is public key. Anything public key today. And so that's why I think we do need to, like...

Leo: Yeah.

Steve: And we know. We know how long it's going to take. It's going to take forever to replace what we have.

Leo: Right. This is a common problem in general for people who cover technology. Fusion's hard.

Steve: Yes.

Leo: Quantum's hard.

Steve: Like the easy problems have been solved.

Leo: Yeah. AI is hard. Self-driving vehicles are hard. Does it mean they're not going to happen? Not necessarily. But maybe not. I think it's good to be skeptical. Dvorak taught me that. He thought everything was crap. But you know what, that's actually, if you're going to pick a default position, that's the most likely correct because most stuff is crap. But then you're going to miss a few jewels, like he thought the mouse was a terrible idea. And, you know, everybody remembers that. Nobody remembers the 101 other things that he thought were crap that went away. So it's something I deal with a lot.

You know, we're talking a lot now about augmented reality. And I think I was right when I said 3D in movies and TVs was a terrible idea and was a gimmick. I'm not sure about - VR I think is a gimmick. AR I'm not sure about. Quantum, you know, the real problem is, as you point out, there's a gold rush because governments are throwing money at scientists. So of course they're going to say, oh, yeah.

Steve: Oh, you know why they are, too. I mean, oh my god.

Leo: Yeah. If somebody does come up with it. I would like, and I think they're starting to, but I would like to see them throw as much money at fusion. Fusion could solve so many of our energy. It'd solve all of our energy issues. Today.

Steve: Yeah, yeah.

Leo: But it's hard. So is it never going to happen? I don't know. We don't know. It's an interesting conundrum for the tech journalist.

Steve: So I have a couple bits of miscellany. Black Hat, the organization, is somewhat notorious for being quite slow to release the materials after their conferences. So an enterprising researcher has put them all up on Google Drive with open public sharing access. He tweeted a link to the collection, and I've captured it in this week's show notes. And to make it easy for those who don't want to track it down in this week's show notes, I also gave it a GRC shortcut of BH, for Black Hat, obviously, BH2022. So if you go to grc.sc/bh2022, that'll bounce you over to Google Drive, and you will see a ton of PDFs containing all of the slides for all of the presentations for Black Hat 2022, which I just thought was cool.

Leo: Yeah, these are great, too. That's great.

Steve: Yeah.

Leo: Thank you, Black Hat.

Steve: Okay. Csurf NPM, the package manager library, has a mistake. So the NPM-hosted JavaScript library named Csurf is an JavaScript library designed to protect applications from Cross-Site Request Forgery attacks, known as CSRFs. You can see where the name Csurf, C-S-U-R-F, got its name. So these are cross-site request forgery attacks. Library protects JavaScript apps from that. Unfortunately, researchers at the security company Fortbridge discovered a CSRF vulnerability in Csurf. And unfortunately, the package apparently cannot be used to protect itself. Since the project's authors have apparently decided it's not worth repairing, they chose instead to mark Csurf as "vulnerable and deprecated." And I suppose, at the same time, as evidence of the need for it. So there.

I didn't think I was going to have anything new to share about SpinRite today. Work is proceeding well. It's running, and I almost have all of the obvious problems fixed, which I created by basically rewriting most of it, except the UI stuff. I may rearrange its real-time activities screen to make better use of its real estate as a consequence of other

things that I've had to change on that screen. You know, lots of field lengths had to change in order to make room for the number of sectors that drives now have and so forth.

But in any event, this morning I encountered a Twitter direct message from Matt Foot, one of our listeners with whom I exchange notes on Twitter. He explained that he was searching for "How SpinRite works," and he encountered a surprisingly recent 2021 YouTube video showing a full demo walkthrough of SpinRite II, which used Roman numerals back then, running on a 20MB Seagate ST-225 drive. I hadn't seen SpinRite II run in quite a while, so I wound up watching the entire 13-minute, well-narrated video. The drive was initially actually in pretty bad shape, but SpinRite fixed it. For anyone who's interested, it's this week's shortcut of the week, so grc.sc/887, which will bounce you over to a 13-minute YouTube video which is sort of interesting. And thanks to Matt.

A little bit of closing the loop with our listeners. Tyson Moore said: "Hi, Steve. Just listened to SN-886." So that was last week. He said: "I have a different take on NIC LEDs" - you know, Network Interface Card or Controller LEDs - "you might find interesting." He said: "I worked for a large Canadian telco that had disabled activity lights on almost all of their switches and routers" - okay, which makes me shudder. But he said - can you imagine that, Leo? Cutting all the lights?

Leo: Must have been very dark in there.

Steve: Oh, my god. Well, you would think the power had failed.

Leo: Right.

Steve: I mean...

Leo: Oh, I use my lights all the time.

Steve: Oh.

Leo: I need those act lights because I need - yeah.'

Steve: Yeah. Yeah. He says: "...from small 1U 16-port access switches to 30U" - that's like 30...

Leo: It's a whole rack.

Steve: 19" rack U height, "...multi-terabit routers."

Leo: Wow.

Steve: He said: "On many devices, this was done through undocumented commands or special firmware."

Leo: Wow.

Steve: He says: "Though it made troubleshooting difficult" - yeah, do you think? - "the idea is that an adversary wouldn't be able to distinguish dormant or lightly used links from busy ones."

Leo: Ah. Okay.

Steve: Okay, "especially when faced with hundreds or thousands of options in a large telephone exchange." He says: "I was never fully sold on the usefulness of this measure, but as a defense-in-depth strategy I figure it couldn't hurt." And I guess my feeling is, if you could like turn them on when you need them or, like, you know, have, like - then, you know...

Leo: Yeah. Because I look at the activity light to know if the printer's talking...

Steve: Oh, I can't imagine not having those little flickering lights.

Leo: I know.

Steve: You absolutely need them.

Leo: And the router and the modem you really need them.

Steve: Yes, to know what the hell is going on.

Leo: Am I connected? Am I not? Yeah.

Steve: Yeah.

Leo: I turn them off, though, on the WiFi access points because I don't like those little green lights all over the house. But other than that, you know.

Steve: That's an interesting point.

Leo: Yeah. I like all those LEDs everywhere.

Steve: Okay. So Tyson, thank you. Also, okay, now, I removed this person's name because I felt I needed to be maybe a little critical, and I didn't want to embarrass him. He said: "Hi, Steve. I enjoyed your discussion of the LastPass breach. Whilst users' vaults are strongly encrypted and therefore presumably fairly safe, do you think there could be another risk in that conceivably the hackers could have introduced a backdoor into the LastPass code? How can we know the code itself is still safe, given that hackers have had access to it for an unspecified amount of time?"

Okay, well, I think this is where the meaning of words matters. Since everyone is on the outside looking in, we don't know precisely what happened, and I doubt we ever will. So we can choose to take their CEO's statement as fact or not. If we do choose to accept it as fact, then exactly what he said matters. He said, this is the CEO who I quoted last week: "We have determined that an unauthorized party gained access to portions of the LastPass development environment through a single compromised developer account and took portions of source code and some proprietary LastPass technical information." Okay, certainly "taking source code" is a world different from "may have modified source code." We must assume that they have well-established source code controls in place, and that verifying the extent of the intrusion would have been their top priority.

Leo: Yeah.

Steve: So again, if we choose to trust the CEO's statement, then there's no danger to us, and we can hope that they will respond to this by making anything like this from ever happening again much less likely. And if we choose not to trust what the CEO said, then nothing he said matters anyway.

Leo: And people may not know that a code repository isn't just a bunch of text files that you can change, and then they just...

Steve: Nobody will notice.

Leo: Nobody will know. There's a whole process, you know.

Steve: Right, right.

Leo: And of course there's a complete log of everything that's been done. They even, I think it's nice, they called the command "blame." And you can see who is to blame for a particular commit.

Steve: Well, and how many times have we talked about like there's a problem, and then they'll go back, and they'll go, oh, that was Johnny when we were having that bad week in 1987 or something. It's like, oh.

Leo: Yeah. Yeah. You can see in any - blame is a git command. But almost in any...

Steve: There is an audit trail.

Leo: Yeah, always you can see which lines were changed, who changed them, when. All of that is completely transparent.

Steve: Right. And on Saturday, Leo, Bill Crahen tweeted: "Hey, Steve. Just set up my Sandsara last night." He said: "Smaller than I imagined, but still fascinating to watch." I don't have mine yet, but that means that they have shipped them.

Leo: Woohoo.

Steve: And presumably I'll have a box on my porch, and you will, too. I think I added, I think I asked for the addition of black sand, and apparently that really freaked out customs for some reasons.

Leo: Yeah, because it looks like gunpowder. I can see why.

Steve: Or like, you know, microcaviar or something. I don't know.

Leo: Can I have yellowcake on mine? I think that would be the best.

Steve: And finally, one of our listeners tweeted: "Just listened to SN-886. I've been in the foo@duck.com beta for about a week now." Meaning the email filtering that we talked about last week. He says: "It has worked as advertised and as you described. I'm not seeing 85% of my email with trackers, more like 50 to 60. It's equally surprising to me who is and who isn't using trackers." And he didn't say more, but now I'm curious, like, oh. I think I need to start routing some stuff through it and find out because that seems really cool to find out.

Okay. And now the one thing I am more excited about than anything else.

Leo: What?

Steve: Yes.

Leo: This must be good.

Steve: I have a new sci-fi author, Leo. Thanks to one of our listeners whose Twitter handle is "Laforge129" - and Laforge, thank you, thank you, thank you - I'm very excited to announce the discovery of a completely new and blessedly prolific science fiction author. This author's name is pronounced Scott Jucha, which is spelled J-U-C-H-A. His website domain is his name, so scottjucha.com.

Okay. So first of all, Scott can write, which is an endangered talent these days. And he has a pleasantly expansive working vocabulary which is a joy to encounter. He writes stories containing well-formed characters who, at every turn, do what you hope they're going to do, and then surprise you by exceeding your expectations. Many of his Amazon reviewers, in their review, like their final review after they've finished the final book in his

The Silver Ships series, state that this is the best space opera series they've ever read. It would take a lot in my mind to compete with Ryk Brown's "Frontiers Saga," but there's room for a top two.

I received Laforge's tweet a week ago today, and since then I've read the first book.

Leo: Oh, I know why you like this. There's 20 of them.

Steve: Yes.

Leo: Oh, my god.

Steve: Actually 24 because he did a short little four-book series that branches off after book 13. So I needed, as I said, Leo, he's prolific.

Leo: Yes.

Steve: He's also on Audible narrated by Grover Gardner.

Leo: Oh, I love Grover Gardner. Oh, good. All right.

Steve: Okay. So, okay. So I read the first book. Then I needed to see what was about to happen next, so I read just enough of the second book to relieve that pressure. Then I went back and reread/skimmed the first third of the first book in order just to relive its key parts because now I knew what the author had in mind. And at one point my wife Lorrie asked where I was in my reading, and I explained that I was re-reading book one. She just shook her head and said: "We are so different that way." And so I explained/asked, I said to her, "Haven't you ever seen a movie that was so good that you immediately watched it again, able to enjoy it even more since you knew how everything fit together?" And that generated some more headshaking.

Leo: I'm thinking Lorrie says life is too short to read more than once. I'm guessing, but I don't know.

Steve: Yeah, exactly. We actually encountered another couple, neighbor friend of ours, out on our evening walk yesterday. And one of them said that, when I was explaining this, that she goes to the end of the book to see how it ends.

Leo: And simplifies it.

Steve: And it's like, oh, my god. Whoa.

Leo: Just cut to the chase.

Steve: So anyway, 10 years ago, in 2012, Scott began writing. In April of last year he finished his 24-novel series which he calls The Silver Ships. All of the first five novels in the series were three times awarded Amazon's #1 Best Selling Sci-Fi book. And they also won the #2 spot twice across multiple science fiction categories of first contact, spaceflight, and alien invasion.

Leo: Well, my three favorite topics. And there's asteroid mining, as well, just to round it out.

Steve: Oh, baby. Actually the main character, Alex, is an asteroid miner in the beginning. So the story is set in the far future. It follows a number of Earth-descended colonies which encounter a very non-human, quite powerful, and quite hostile alien race. Ever since I caught up with Ryk Brown's Frontiers Saga series, I've been casting about, looking for something next, while Ryk continues working on his books. So I am absolutely certain that I've found what I've been looking for. I have no idea what's going to happen, but I love what has happened so far.

Leo: Oh, boy.

Steve: The books are all on Amazon. They're all available through their Kindle Unlimited program and, as I mentioned, also on Audible, narrated by Grover Gardner. Look for The Silver Ships series.

Leo: Adding it to my list.

Steve: And oh, Leo, oh.

Leo: I like it that it's Kindle Unlimited because that means I can dip into it and just see if I like it.

Steve: Yes.

Leo: Good.

Steve: I already told JammerB. John is in the middle of a trilogy, but he can't wait.

Leo: I'm just finishing "The Singularity Trap," so I've got a ways to go.

Steve: Yeah.

Leo: Only a few more chapters on that.

Steve: Okay. Let's take our last break.

Leo: Yes.

Steve: And then we're going to talk about what Symantec found in more than 1,800 mobile apps.

Leo: Wow. Have you ever attempted to write a sci-fi novel yourself?

Steve: It'd just be a bad use of my time.

Leo: Yeah, no, it wouldn't be now. Maybe in retirement, whenever that is. I wish I could because I would love to be a writer. That seems like the perfect occupation; right? It's kind of like coding; right?

Steve: I would love it, too.

Leo: Yeah.

Steve: I mean, if I - yeah.

Leo: But I can never think of any good stories, so I don't think it's going to happen.

Steve: Apparently Lorrie is listening to us.

Leo: Uh-oh.

Steve: Because she sent me a text, and she said that she had watched some movies multiple times, she said, the Harry Potter movies. She said, "Big fan of the Harry Potter movies."

Leo: There you go.

Steve: And for me, I think probably "The Matrix," I'm sure I watched it a second time. "Terminator," the first time I saw that I would have had to, like, watch it again. So, you know, there are just some that are really good.

Leo: All right. Yeah, I'm not a big movie re-watcher. I try because I foolishly have bought movies in the past, and that's a waste if you're not going to watch it again.

Steve: That's a good point, yeah.

Leo: Yeah.

Steve: Okay. Embedding AWS Credentials. Last Thursday, Kevin Watkins, a security researcher with Symantec, revealed the results of Symantec's sobering research into a previously unappreciated, or at least grossly under-appreciated, serious weakness and vulnerability created by the way today's increasingly powerful mobile applications are being developed. The problem surrounds the collision of the increasingly ubiquitous use of "The Cloud" by mobile apps with the merging of out-sourced code libraries and SDKs which use "The Cloud" to contain their sometimes massive databases. Another problem appears to arise from a failure to follow the old adage which carries the well-known abbreviation: RTFM. We know what that stands for.

The data belonging to both users and the enterprises hosting these dangerous ill-designed apps are thus put at risk, as is the data of all the customers of these enterprises. The problem is big. So I wanted to get specific and put a sharp point on this, fleshing it out by sharing what Symantec's research revealed. They open this report with a punchline: "Over three-quarters of the apps Symantec analyzed contained valid AWS access tokens that allowed access to private AWS cloud services." Okay. Now, that was 1,859 iOS and Android apps which were found to be leaking actionable AWS cloud credentials that MUST be kept private.

So here's how Symantec framed the problem and explained what they found. They said: "Most of us by now have been impacted in some way by supply chain issues. An increase in the price of fuel and other items, delivery delays, and a lack of product availability are just some of the consequences of supply chain issues stemming from recent events around the world." Okay, well, that's not - doesn't apply to us.

They said: "However, in the context of software and technology infrastructure, the consequences resulting from supply chain issues are very different. Mobile apps, for example, can contain vulnerabilities introduced in the supply chain that can potentially lead to the exposure of sensitive information, which in turn could be used by threat actors for other attacks." They said: "Mobile app supply chain vulnerabilities are often added by app developers, both knowingly and unknowingly, who are likely unaware of the downstream security impacts putting not only the app users' privacy at risk, but sometimes putting their company and employer's privacy and data at risk, too."

Okay. So in other words, this is what I would call the "modern modular software component assembly dilemma," where it's too easy to plug this library into that library and have this API calling into that API, and where everything just appears to work, but without the developers ever obtaining a full in-depth working understanding of exactly what's going on. And of course that's the whole point of using a plug-in modular library and its APIs is that you don't need to learn everything about what the serving library is doing. The trouble is, this is also the way implementation mistakes happen. And in the case of AWS, the mistakes have huge consequences.

So they said: "Similar to the supply chain for material goods, mobile application software development undergoes a process that includes the collection of materials, such as software libraries and software development kits (SDKs); manufacturing or developing the mobile application; and shipping the end result to the customer, often using mobile app stores. This research," they said, "examined the type of upstream supply chain issues that can make their way into mobile apps, making them vulnerable. The issues include mobile app developers unknowingly using vulnerable external software libraries and SDKs; companies outsourcing the development of their mobile apps, which then end up with vulnerabilities that put them at risk; and companies, often large ones, developing multiple apps across teams, using cross-team vulnerable libraries in their apps."

They said: "In order to better understand the prevalence and scope of these supply chain vulnerabilities, we took a look at publicly available apps in our global app collection that contained hard-coded Amazon Web Services credentials. Hard-coded cloud credentials is a type of vulnerability," they said, "we've been looking at for years and have extensively covered in the past. This time, in order to get to the bottom of the supply chain impacts caused by the issue, we've looked into why app developers hard-code cloud credentials inside apps; where the hard-coded credentials are located in the apps tracking the sequence, or chain of events leading to the vulnerability; and, finally, the size of the problem and its impact."

They said: "We identified 1,859 publicly available apps, both Android and iOS, containing hard-coded AWS credentials. Almost all were iOS apps, 98%" - which is really a curious number - and, they said, "a trend and difference between the platforms we've been tracking for years, possibly linked to different app store vetting practices and policies." And I don't understand, or maybe the application market is just that different between iOS and Android.

Leo: Oh, it is, it is.

Steve: Yeah. That would be my guess. And they said: "In any case, we examined the scope and extent of the risks involved when AWS credentials were found embedded inside apps. We found the following: Over three-quarters (77%) of the apps contained valid AWS access tokens allowing access to private AWS cloud services."

Leo: Geez, Louise.

Steve: I know, Leo.

Leo: They're like hard-coded in?

Steve: Yes. Yes, it's like, you know, we've talked about how lame it is for routers to hard-code in a username and password for like some backdoor. Cisco spent years recovering from that practice.

Leo: Is this because like they're using AWS as a server for images or something, and so they have to do that?

Steve: That may be one of the reasons.

Leo: Maybe it's not private stuff, it's like parts of the app.

Steve: Well, yeah, but this is access to private AWS cloud services.

Leo: Oh. Seems like a bad idea.

Steve: So sounds bad. And then they also said: "Close to half (47%) of those apps contained valid AWS tokens that also gave full access to numerous, often millions, of private files via the Amazon Simple Storage Service."

Leo: Okay, that's definitely bad.

Steve: That's definitely bad. And they're going to get much more specific about this in a minute. So then they said: "We will explore the type of private data exposed in the examples discussed later in this blog, but the message is clear: Apps with hard-coded AWS access tokens are a vulnerable, active, and present risk." They said, okay, well, and present a serious risk, they said.

So I should explain that it would be entirely possible for apps not to embed - and Leo, you already get this - not to embed static AWS access tokens into their code. And again, how many times have we talked about the insanity of a Cisco router, for example, embedding some backdoor access username and password into its firmware where it's ripe for discovery? It's just malpractice and laziness. In the case of well-connected mobile apps, it would be trivial to have apps reach out to obtain the AWS token on the fly over a secure encrypted and authenticated connection. That would have the added flexibility of allowing the app's developers to change AWS credentials on the fly, if some access right problems, such as we'll be discussing in a minute, were to be found.

In any event, Symantec continues. They said: "We then looked into why and where exactly the AWS access tokens were inside the apps, and if they were found in other apps. We discovered" - get this - "that over half (53%) of the apps were using the same AWS access tokens found in other apps. Interestingly, these apps were often from different app developers and companies. This pointed to an upstream supply chain vulnerability, and that's exactly what we found," they wrote. "The AWS access tokens could be traced to a shared library, third-party SDK, or other shared component used in developing the apps.

"As for the remaining question of why app developers are using hard-coded access keys" - Leo, to your point - they said: "We found the reasons to include downloading or uploading assets and resources required for the app, usually large media files, recordings, or images; accessing configuration files for the app and/or registering the device and collecting device information, storing it in the cloud; accessing cloud services that require authentication, such as translation services, for example; or no specific reason, dead code, and/or used for testing and never removed."

They said: "If an access key only has permission to access a specific cloud service or asset, for example accessing public image files from the corporate Amazon S3 service, the impact may be minimal. Some app developers may be assuming this is the case when they embed and use hard-coded AWS access tokens to access a single bucket or file in Amazon S3. The problem is often that the same AWS access token exposes all files and buckets in the Amazon S3 cloud, often corporate files, infrastructure files and components, database backups, et cetera. Not to mention cloud services beyond Amazon S3 that are accessible using the same AWS access token."

They said: "Imagine a business-to-business company providing access to its service using a third-party SDK and embedding an AWS hard-coded access key, exposing not only the private data of the app using the third-party SDK, but also the private data of all apps using the third-party component. Unfortunately, this is not an uncommon occurrence, as you can see in the following case study examples."

Okay. So we've got I think three here, and they kept them anonymous, not to embarrass the actual provider. But these are specific iOS apps. They said: "We found, in an Intranet platform SDK, we found a business-to-business company providing an Intranet and communication platform that had also provided a mobile SDK that its customers could use to access the platform. Unfortunately, the SDK also contained the business-to-business company's cloud infrastructure keys..."

Leo: Oh, please.

Steve: Oh, "...exposing all of its customers' private data on the business-to-business company's platform. Their customers' corporate data, financial records, and employees' private data was exposed. All the files the company used on its Intranet for over 15,000 medium- to large-sized companies were also exposed. Why did the company hard-code the AWS access token? In order to access..."

Leo: Because they didn't know better.

Steve: Exactly, yes. "In order to access the AWS translation service."

Leo: What? Oh, please. Oh, please.

Steve: "Instead of limiting the hard-coded access token for use with the translation cloud service, anyone with the token had full unfettered access to all the business-to-business company's AWS cloud services and uses."

Leo: Wow. I'm starting to think that, just as we license drivers before they're allowed to go on the road, we should have some sort of minimum competency standard for programmers before they're allowed to publish software.

Steve: I know.

Leo: I mean...

Steve: I know, and it's completely lacking. Doctors and lawyers have to go through extra school and get certified and pass tests to demonstrate that they have, in the case of lawyers, a basic understanding of the way to do their job. Programmers, not so much.

Leo: Wow.

Steve: I mean, not at all. And, you know, there are, as we know, ITProTV produces certifications. Those exist. But you don't have to have one in order to write code.

Leo: Yeah.

Steve: We have another instance, a digital identity and authentication. They said: "We discovered several popular banking apps on iOS that rely on the same vulnerable third-party AI Digital Identity SDK. Outsourcing the digital identity and authentication component of an app is a common development pattern as the complexities of providing different forms of authentication, maintaining the secure infrastructure, and accessing and managing the identities can incur a high cost and requires expertise in order to do it right. Unfortunately, in this case, things were not done right.

"Embedded in the SDK" - which, again, was shared by several popular banking apps on iOS. "Embedded in the SDK were cloud credentials that could place entire infrastructures at risk. The credentials could expose private authentication data and keys belonging to every banking and financial app using the SDK. Furthermore, users' biometric digital fingerprints used for authentication, along with users' personal data (names, dates of birth, et cetera), were all exposed in the cloud.

"In addition, the access key exposed the infrastructure server and blueprints, including the API source code and AI models, used for the mobile operation. In total, over 300,000 biometric digital fingerprints were leaked across five mobile banking apps" - five mobile banking apps - "using the SDK."

And finally, online gaming. They said: "Often, already established companies rely on outsourcing, or partnering, with other business-to-business companies for their digital and online services. This allows them to quickly move their brand online without having to build and support the underlying technology platform. At the same time, by relying on the outsourced company to run the technology platform, they often have to give exclusive access to their business data. Furthermore, they have to trust that the outsourced company will protect the online private data, not to mention the reputation of the brand overall." Boy, talk about skating on thin ice.

They said: "We found a large hospitality and entertainment company depending on another company for their technology platform, even forming a sports betting joint venture with the company. With a highly regulated sports betting market, the complexities of building and supporting infrastructure for online gambling cannot be underestimated. Unfortunately, by giving the joint venture company exclusive access to that part of its business, the company also exposed its gaming operations, business data, and customer data to the world.

"In total, 16 different online gambling apps using the vulnerable library exposed full infrastructure and cloud services across all AWS cloud services with full read/write root account credentials." And they said: "All of the organizations whose vulnerable apps were discussed in these case studies have been notified about the issues we uncovered."

Leo: Now you know why there are so many S3 bucket exploits.

Steve: Yes.

Leo: I mean, this happens, this is probably the most common kind of breach.

Steve: Yes.

Leo: It seems like Amazon maybe isn't doing a good job of explaining permissions or something. Like maybe Amazon can fix this.

Steve: Yeah. I agree with you. It is - I think what happens is that it's so easy to use Amazon S3 buckets. I have and maintain a bunch of my own, like every time I upload the edited audio to Elaine, copies of the podcast go into an S3 bucket, just to have it in the cloud so that it's archived archived. The problem is I think that developers get it going, like without any access controls, and then they just forget to go back and lock it down and, like, restrict it so that it's not just left as public because, unfortunately, that's the way it is by default. It's public unless you make it private. And that seems to be the mistake that's being made. Crazy.

Leo: Wow.

Steve: And it's also, maybe it's because it's remoted that somehow you don't have the same level of visibility into access controls that you do for something that is local. Or just they didn't RTFM.

Leo: Yeah. I mean, I seem to remember having set up quite a few of these. They kind of - this is like the password. This gives you access. They even hide it; right? You have to reveal it. It seems to me that - I don't know what's going on. Lazy.

Steve: It's not good.

Leo: Or, yeah, or something.

Steve: Well, and the problem is, and this was the point that Symantec was making, is that we've got a difficult, clearly a difficult-to-secure or too often not secured cloud resource, which because of the way business to business are now beginning to contract with each other for big chunks of responsibility, that cross-business contracting is what Symantec is calling a new form of supply chain. And if your provider of these services isn't careful with the design of their product, its inherent security appears to be lacking. There's no other way to explain why 16 different, entirely separate gambling environments which all use a common package, would have had all of their data cross-exposed if it weren't for the mis-design of that package.

Leo: Yeah. Unbelievable. Yeah, that's the other problem. People just kind of willy-nilly just pasting packages in there, yeah.

Steve: Yeah. It's like, oh. Well, and because who wants to learn all that.

Leo: I don't want to write all that code. Oh, look, somebody wrote it.

Steve: We can get it from Joe. Great.

Leo: Yeah, yeah.

Steve: Import. Import. Press the Import button.

Leo: Yeah. Well, that, I'm kind of sympathetic to that. It's really up to Joe to make that secure. Joe's got to do a better job. Wow. Well, another great Security Now!, as always. Thank you, Steve. Steve Gibson lives at GRC.com. There's all sorts of great stuff there. Of course SpinRite, the world's finest hard drive recovery and mass storage recovery and maintenance utility. And you can pick up a copy right now, 6.0. 6.1 is in process, will be out sometime, and you'll get it for free if you buy today. And he's working on 7.0, apparently, I just found out. So, busy man.

Steve: Plans. Plans for 7.0.

Leo: Planning. You've got to think about it first.

Steve: I know what it's going to be.

Leo: You've got to know what S3 buckets to stick it in.

Steve: That's right.

Leo: That's one good thing about assembly language. Probably not a lot of S3 tokens.

Steve: Hopefully it's not a leaky bucket.

Leo: Yes.

Steve: And there are no one else's modules that I'm importing.

Leo: Nobody else makes them. Right.

Steve: That's right.

Leo: Where are you going to go to find those modules? Look, we've got a whole library of assembly language routines you can use. Aggressively non-cross-platform. That's Steve. Right there. In a nutshell. Nope, 8086, that's it. But, boy, I tell you what, it works good. He also has of course the copies of this show. Now, he has a 64Kb audio, which is kind of the standard audio podcast version. But he also has a 16Kb version for people who really don't want to spend the bandwidth. And he's got transcripts. Elaine Farris writes those out after every show, takes a few days. That's really nice to read along as you listen or to use for reference or to search. I bet you, I know somewhere out there there's somebody with three shelves' worth of three-ring binders with all the transcripts printed out and indexed. Don't you think, Steve? Somebody's done that.

Steve: Yeah.

Leo: The Magnum Opus. All of that's free at GRC.com. All you've got to do is buy SpinRite. Everything else takes care of itself. Lots of other stuff, as well. You can leave him a question or a comment or suggestion at GRC.com/feedback. Probably the better way to do it is to go to his Twitter account, @SGgrc. You can DM him there. His DMs are open. After the fact we have the show, as well, at our website, TWiT.tv/sn. We've got the 64Kb audio, and we also have video, which is weird, but we've got it. [Beeping] Oh, time to go. Show's over. I heard, I swear, and I must be dreaming, wind chimes. Do you have wind chimes in your studio, in your office?

Steve: That was Lorrie's iMessage coming in. Yes, because I do have sounds associated with everything.

Leo: Nice. So she's very relaxing wind chimes.

Steve: Nice sound, yeah.

Leo: I thought I must have died and gone to heaven. Steve Gibson has wind chimes at the GRC Labs.

Steve: There's no wind here except what comes out of my mouth.

Leo: A little hot air, but no wind, yeah. Hot air here today, I'll tell you. We also invite you to watch live, if you want. You could. We stream it live as we do with all our shows. We just open up the cameras in the studio and let you watch at live.twit.tv. I's usually right after MacBreak Weekly, that's about 1:30 Pacific, 4:30 Eastern, 20:30 UTC. If you're watching live, you can chat with us live in IRC. That IRC site is irc.twit.tv. You can go there with a browser. It has the deets on how you could use an IRC client if you prefer. If you do it more than once, you should probably get an IRC client. Of course the Discord folks are also chatting behind the scenes, if you're a member of Club TWiT.

YouTube Channel also, dedicated to Security Now!. That's most useful, I think, well, if you watch a lot of YouTube it's great. But also it's a great way to send just a little clip to somebody because that's easy to do on YouTube. So if you heard something you thought, I've got to send this to Joey, he keeps putting our S3 bucket credentials in the app, you could just snip that part, send it to Joey, let Steve explain why it's a bad idea. Oh, of course the best way to get it would be subscribe. That way you can add, to your six-foot shelf of binders, you could add the complete Security Now!. We are now in our 18th year. 19th year. 18th year.

Steve: 18th, yeah.

Leo: Episode 887. So collect all 887. We only do 10 at a time in the feed. So if you want to go past 877 you'll have to go to the website, either Steve's or ours, and download all of them. But they're all up there on the website. You can get them all.

Steve, have a great week. I am going to be reading this new Silver Ships. I'm excited about it.

Steve: Oh, I think you're going to like it. It's just so pleasant.