CR064

New vulnerabilities in the digital age with Jeremy Snyder Founder of FireTail

Capgemini

CR064

New vulnerabilities in the digital age with Jeremy Snyder Founder of FireTail

Disclaimer: Please be aware that this transcript from the Cloud Realities podcast has been automatically generated, so errors may occur.

[00:00:00] Oh You know, this is perfect because you know what day it is today. What day is it today? No. It is World Password Day, Rob. It is. Oh my word. This is exciting.

Welcome to Cloud Realities, a conversation show exploring the practical and exciting alternate realities that can be unleashed through cloud driven transformation. I'm David Chapman and I'm Rob Kernahan and unfortunately Sjoukje is not with us this week. She is not feeling well, so get well soon, Sjoukje!.

And in this episode of the show, we are going to dive into the world of API security. Much like software driven infrastructures, the world of security and cyber has also moved from being infrastructurally concerning around things like firewalls [00:01:00] and physical infrastructures, much more up into the application layer and the massive complexity of loosely coupled microservices and APIs.

But before we get to that, Rob, as I was walking into the office the other day. Now we just for a reminder of our listeners who don't remember this I didn't hear the episode Rob has got a side hack where when we have the perfume sellers that are in the reception Rob does a couple of shifts on there just to you know Just to square away a little bit of extra for the older for the old pension bring forward that retirement date You know every penny counts But guess what?

I saw in reception the other day Rob and I'm thinking Rob was gonna want a slice of this What was that Dave? fruit. Fruit. Yeah. So I'm now going to become a green grocer. Yeah. Yeah. So this dude's there selling bananas and grapes. So you don't have to get that complicated with it. It's not like you need a dense supply chain.

Bananas [00:02:00] and grapes sold at hugely inflated prices. And I bet you he does quite a roaring trade as well. Repositioned as a healthy, you know, like a healthy office energy snack. I think, you think about it, because literally where we are, there's like shops around the corner that will just sell you that at reasonable shopping prices.

That's just buying a banana, that's not buying a healthy office energy snack, is it? Well no, indeed. It's all marketing, isn't it? That's quite clever marketing, isn't it? Exactly. You never thought about grapes as being healthy, or a banana as being able to give you an energy boost, so. Exactly. I didn't see, I didn't see, I didn't see it repositioned and reframed as that.

But, but now I've seen it. I can't, I can't unsee it. So you're basically saying that I should promote a business model with gullible people associated. Well, yes, that would work. And that's also, you could reposition that in the multiple different areas, but I was thinking more like apples. Well, it's definitely diversification.

So I'll give you that. I thought I'd bring a tip this week before I said, you know, what are you confused about this [00:03:00] week? Well, so apart from the, what's my business plan for the future for selling high energy, healthy snacks. In disguise was asynchronous communication and the stress it causes in our workplace.

And I'm a bit about, I'm a bit about this where it's like, let's take the three dots as the example, right? Those three little bouncy dots when they first arrived, they were amazing. Oh, I've sent an asynchronous message and somebody is responding. What it turns out to be is it causes massive stress because you've had that thing where the three little dots start bouncing.

And then they stopped bouncing and then the three little dots come back and then they go away again. And then there might be a message or not. And basically what that means is you're sat there staring at the async interface on the messaging, making it synchronous

essentially. Is that particularly worrying if you've sent like say a punchy one?

Yeah. Yeah. You got a bunch of you on and then the dot start and stop and start and stop and start and stop. And you think, Oh God, they've deleted it. Redone. And it's the whole, and then there's the etiquette point around, uh, people who just say hi and then you respond with hello and then they [00:04:00] don't ever respond again.

Right. And there's actually an etiquette online where you're supposed to say hi and then a little dash and put your request in. So it's like a fully formed thing, but if you think about how passive aggressive they just saying hi is and then not responding, and I get the timing and thing and everything else, but it just causes stress.

And I'm thinking actually. Is it, is it a good thing or a bad thing? And I think it probably could be a good thing if everybody affects their behavior, but I fear that it's not, and it's not going away, and it's going to get worse, and I'm wondering where it's going to go next. But I don't actually think it helps a lot, because it just gives you more interruption, causes you stress, sucks you in, and then you get more frustrated with it.

You know what the other minefield is here? Are you confused? There's a whole other minefield that's connected to this, that you haven't even begun to unpack. And it is the demographic generations and the use of different emoticons to mean different things. Oh, yeah. Apparently the thumbs up. The thumbs up is exactly where I was going to go.

You're not supposed to use it anymore, are you? Apparently for a Gen Z, getting a thumbs up is like getting the middle finger. [00:05:00] I was surprised at that as well. I thought, you know, there I am, Gen Xer, right at the end, um, uh, thumbs up. That felt like a good thing. Happy go lucky. It feels happy go lucky. Yeah, yeah.

It's like, that's good, isn't it? Yeah. Yeah, but apparently not. Yeah, I saw that as well. So there's this minefield of, of, uh, it's, it's passive aggressive. It's frustrating. And then absolutely you appear to be offending people when you think you're being nice to them. Well, thumbs up to that Rob. I'm not sure we're gonna, I'm not sure we're going to crack that one.

That's just when the new demographic generation alpha comes in, it's going to get even worse. going to get even worse. Anyway, look onto the main subject of the show today. We talk about cyber quite a lot on the show. It is a constantly evolving world. The more sophisticated our technology gets, the more sophisticated cyber gets, the more sophisticated the attacks get.

And that is absolutely true in the cloud. When you have software driven architectures. Delighted to say that Jeremy Snyder, the founder of FireTail, is joining us today to discuss this. Jeremy, hi there. Good to see you. Thank you for making time for this. You just want to [00:06:00] introduce yourself and say a word or two about vital.

Yeah, happy to. And thanks so much for having me. Real pleasure to be here. I'm Jeremy. I'm one of the co founders here. I am more on the business side. My partner Riley is more on the technology side. But you know, my journey in it has been about 20 Six, seven years at this point, I started as a hands on keyboard IT and cyber practitioner for a couple of SAS and software companies in the late 90s and then a video game company for a while.

A fascinating experience. I would never go back into the video game world. If you see how the sausage is made, it's not nearly as glamorous as you might think. And it is probably the most high stress cutthroat business environment that I've ever worked in. But, uh, it was a fun experience to go through.

And then I shifted into the cloud and more into customer facing roles and kind of, let's call it sales, sales, engineering, evangelism, things like that. And I've spent the last seven ish years [00:07:00] at this point, uh, eight years, actually at this point in You know, kind of hands on cloud security, working with customers around the world on large scale cloud security initiatives.

And then the last two years really focused on APISecurity. We started fire tail in February of 2022 to really try to solve security for APIs. We always had the vision that what we wanted to create was effective software that actually makes the problem less of a problem, you know, kind of actually really provides value as opposed to providing flashing lights around a problem.

So why don't we start with the world of new technologies and with new technologies comes vulnerabilities and vulnerabilities that previously weren't there, I think. So I think if I'm right in saying Jeremy, FireTail was born out of Effectively that observation and therefore, like, what do we do about that?

Yeah, absolutely. And I mean, do you want to hear the [00:08:00] story of how FireTail was born and what the observations were? Yeah, please. Yeah. Yeah. Look, I've been working in the cloud since 2010. I joined what was at the time a little startup called AWS. They might have heard of them. Where did they go? Where did they go?

Yeah. I don't know. I feel like they might still be a thing. But anyway. You know, Marcel, can you quickly Google that?

I will say at the time it was a very different place in. the technology landscape. We spent all our time running around trying to convince CIOs and CTOs that this thing was real and also trying to convince them that they actually had security and privacy and all of these concerns that you would hear about at the time.

And by the way, one of the main questions that we got asked time and again was, Oh, so Amazon bought too many servers and you're trying to rent us spare capacity. And you know, we got that in four out of five conversations at the time. But, you know, I've kind of stayed in the cloud ecosystem ever since, and I will say around the late [00:09:00] 20 teens, we started to see this shift in what I would call a movement from cloud 1.

0 to cloud 2. 0. And if you think about it this way, cloud 1. 0 is just the fact that, hey, there is this cloud environment, whether it's AWS, Azure, GCP, some of the other players in the space, whatever. But we're really just using infrastructure the same way we've always used infrastructure. You know, so a very common exercise that you would see at the time was customers looking at a migration, they would literally go through their racks and say, Okay, well, I've got X servers with these CPUs and this much memory and this much disk space, and I'm just going to go over to, you know, the Azure catalog and select roughly the same virtual machines, and it's Copy paste, right?

And you'll hear lift and shift. You'll hear all different paradigms or descriptions applied to that. But inevitably, what ends up happening for organizations over time is that they realize their bills are higher than they expected them to be. [00:10:00] And secondly, they start to ask themselves that question of, well, you know, I was promised all this flexibility and agility and so on.

And I'm not really seeing it right now. Right, right, right. Absolutely. I recognize that. Yeah. And it's nothing to do with the cloud platform or the technology platform. It's the way that they operate. And when they come to that realization, then they start to say, well, are there

other ways that we could be running our workloads that might give us more flexibility and agility?

And that is that kind of shift that I talked about from 1. 0 to 2. 0 on the cloud side. And really what that involved for most of the customers that I was working with at the time. was getting rid of virtual machines. Maybe they didn't get rid of all of them, but they start to look at their workloads and look at ways that they can re architect them or refactor them and move more towards containerization, move more towards serverless platforms, etc.

And it was that shift that created this huge number of APIs. And it actually [00:11:00] started to drive organizations to look at an application and say, well, you know, to use the classical buzzword, I've got a monolith over here and it you know, processes 100 different functions, I should actually have 100 different little function applications that all talk to each other over API's.

And that was the shift that we saw in, let's call it like 2018 to the 2020 timeframe that led us to think about, well, what comes out of that transformation, a huge number of API's. And then what do those API's represent? Well, they're this connection point, but they're also an attack surface. And that is, you know, the inspiration for fire tail.

I see. And just to give us a little bit of texture around that, was it you by yourself? Was it you and a colleague? And was it one of those situations you were sitting around having beers or coffee and chatting around these things? What was the actual moment of inspiration? Uh, well, it was me and a former customer of ours, and we were watching the same [00:12:00] transformation from different sides of the table.

Me from the software vendor side and him from the practitioner side. And it so happened that that cloud security company I was with got acquired about two months into lockdown and over the course of lockdown, as I was kind of completing the retention period with the company that acquired us, we, you know, we're just staying in touch.

Uh, over WhatsApp and having calls from time to time and just kind of catching up on personal, you know, what you're working on. And then when I left that company, then we reconnected and we started talking, uh, mostly over WhatsApp about, well, what are you going to do next? Well, what are you going to do next?

And it so happened that. He had left that organization where he was a customer of mine and gone on to another startup that wasn't really working out too well. And so he was also thinking about what he was going to do next. And, you know, we said, You remember, we were talking about that whole transformation and that whole, you know, growth in API's, etc.

What do you think about that? And then, by the way, for you as a software [00:13:00] developer, what do you do to provide security around your APIs? And then we kind of realized that there wasn't a lot of attention being paid to that space. And, uh, you know, that was really kind of the all right, let's go try to figure something out together.

Well, awesome. Let's digress a little bit before we dig into what that attack surface looks like and then, you know, kind of the new approaches that you can take to, to protect against, uh, attack of that. Let's talk at the beginning, though, for those who aren't familiar with the world of, uh, Security detection and approaches within the world of security.

Take us through what the traditional approaches to dealing with security were pre this web 2. 0 world that you describe. And perhaps why they're not effective in this new world. Yeah, look, I think there's kind of a few different ways that problems like this would have been attacked in the past. One of them is through simple network observation and kind of

network monitoring.

And there's [00:14:00] Thousands of tools that have existed for decades around network monitoring. And the reason that a lot of those approaches aren't effective for API security is that when we started to dig into This research project that we did where we tried to kind of, uh, capture a list of all of the API breaches that we had seen.

One of the common threads was something like 80 or 90 percent of the breach incidents were people who were actually authenticated on the platform, but then were misusing their access to access data that they weren't authorized to access. Why is this a problem? Well, this is a problem because from a network perspective, what the logs will show you is a bunch of normal traffic.

And so you're not going to necessarily understand kind of the layers of context that you need to say that, like, okay, this is Jeremy. He's logged in. It looks all normal. Jeremy is now accessing Marcel and Rob's data. You know, the network logs just don't have to be right on top of that. [00:15:00] Jeremy Rob would be right on top of that.

Are you? So, but you're saying I can breach Marcel pretty easily? Is that absolutely no problem? We breach him weekly. He's famously owned in the IT sphere. He just doesn't know it. Basically, you know, he's just on the internet. That's the way we describe Marcel. But he gets by, he gets by. He's really catching a lot of stray bullets here in this conversation, poor guy.

You know what though, deep down he does deserve it. So we're all good with it. I'll leave it to you guys to sort that out. But, but, you know, on the network monitoring side, that was really one of the core things. And then one of the other, let's say, common traditional approaches would be to install a bunch of endpoint agents.

And these could be, you know, your classic anti malware or your endpoint detection and response, etc. The real challenge around those is that again, if you think about the compute infrastructure that customers are using as they make this transition, it's a bunch of serverless functions and it's a bunch of containers.[00:16:00]

And a, a lot of those endpoint agents don't work on those platforms. B, these platforms are now super ephemeral. And there's a huge overhead to installing the agents and deploying them and making them connect and talk back and phone home and everything. And that investment, when a lot of security practitioners we talked to about this, they're like, yeah, why would I go through the trouble of trying to package an agent into a container that's going to live for eight hours?

You know, the, the, the juice is not worth the squeeze, so to speak, right? And so, like, we looked at both of those approaches and kind of quickly discounted them, or, or let's say not a hundred percent discounted, but realize the shortcomings that they have. I think you're calling out really interesting point was because it's security used to be walled garden, stick it in our data center and it's behind a firewall and they're in palatable, aren't they?

And the, the, the illusion of walls. Well, and then cloud came along and cloud native loosely coupled, highly cohesive architectures and security didn't catch up. So the thinking around how you, uh, protect loosely [00:17:00] coupled, highly cohesive architectures one. And then you say the ephemeral nature and state. So things move around a lot.

I think it feels like and it still does a little bit that many security organizations got confused by that and didn't know how to respond because they were used to locking down something that was quite static. And then suddenly everything started moving and they just went, I

don’t know how to cope.

Yeah. And I mean, look, that’s why you saw a whole rise of the category of company that I used to be with, you know, the cloud security posture management. Now, I think, you know, CNAP or cloud native application protection platform is the buzzword that would get thrown around that. But to your point, it really is Okay, we don’t know how to think about this whole set of things.

And by the way, we don’t even know where the security risks are on this newfangled cloud platform that you’re talking about. And that created a space for these companies who are really, you know, If you boil it down, it’s actually a pretty simple set of capabilities, right? It’s fetch a list of all of the resources being [00:18:00] consumed and their configurations, and then run that against a rule set that analyzes what is a good configuration, what is a bad configuration.

And You know, it’s a very simple use case, but it’s a use case that your point wasn’t widely understood before and was something that a lot of security teams were really struggling to keep up with. So one of the things in the traditional world and probably carried over, and I’m sure is done in the in the cloud world today by a number of enterprises was the.

Penetration test. So, you know, the traditional approach is to, you know, kind of bombard an organization from the outside and then look at what breaks on then, you know, make a report of that. That doesn’t function here either. What’s the issue with something like that in this circumstance? The only issue is that it’s only as good as your pen testing team is.

I’ll give you an example just yesterday, a vendor that we actually work with who will remain nameless, but they gave us their report about their audits, etc. And in their reports, they say, yeah, we do [00:19:00] twice annual pen testing. And their pen testers aren’t very API savvy. So we discovered an issue in about 30 minutes that their pen testing, which has been going on for a number of years, twice a year, had never uncovered, and it was the so called broken object level authorization.

It was again that case where a logged in user with You know, very simple changes to a couple characters in a query could access somebody else’s data via API. And, you know, I think if you’ve got good pen testers who really understand the nature of the application that you’ve built and the technologies that you’re using and the risks around those technologies, pen tests are hugely valuable.

But that’s just not consistent. And that’s the real challenge. And that’s, again, a hangover from once I’m through a particular perimeter allowing access to multiple resources. So, authentication has occurred but you don’t have a granular enough authorization to say should you be accessing this resource downstream, etc.

And it’s [00:20:00] that mixing of old world cultures with new. We’ve arrived here in the architecture, therefore we must be safe and it’s a very old way of thinking about security posture. Yeah, absolutely. In the world of APIs, there’s this kind of authorization triangle that gets talked about. And it is principle, resource, action.

So it’s, you know, principle, who is it that is making the request? Resource, what is the data they are making the request on? And action, what are they trying to do with that data? So that could be as simple as Jeremy viewing Jeremy’s profile. Or that could be Jeremy editing Jeremy’s profile. All of those sound great.

Jeremy viewing Rob’s profile? Maybe okay. Jeremy editing Rob’s profile? Maybe not. And so you see there’s like three variables in there that you have to take into account. And strictly

speaking, kind of a zero trust model would be that unless all three map to yes, The answer should be no. And unfortunately, that [00:21:00] is not the way that most APIs are written today.

Isn't it like analogous to maybe I'm oversimplifying this. So do feel free to It's not like you Dave to oversimplify something. I try my best Rob every day. You know, the world of say infrastructure and applications, when you looked at things like security, it was predominantly there was a lot of it in the In the infrastructure layer exactly to Rob's point.

It was about the walled garden. It was about firewalls. It was about protection at that level. And then, you know, a directory service and things like that. And yeah, there was a certain there was a certain infrastructural skill set that went along with that. And as we've gone to the cloud, not only is the skill set You know, for traditional infrastructure operations had to become more like development and the world of application development, you know, to drive platforms and to drive software driven infrastructures and things like that.

Is it true to say that this is analogous in some way to what you're describing in the shift in security? Yeah, very much so. And I mean, just to kind of dive into that analogy, [00:22:00] I would say historically, the way that many security organizations thought about it was we've got this perimeter layer of defense.

Once you're inside the perimeter, if you can kind of, in a valid method, get through that perimeter, or even, you know, through an attack, get through that perimeter, once you're inside, we kind of assume you're okay to go do where you want, go where you want to go, do what you want to do. And the challenge to your point is that in this cloud world, where is your perimeter?

Like, you can throw up a few layers of network infrastructure and whatnot on the cloud side, but there are, you know, in many cases, there's one configuration change that takes something that is behind three layers of protection, and it's all of a sudden public. I'll give you an example on this point. You know, there is a one configuration setting that you can make on backups, on EC2 instances, on AWS.

AWS. That will make a backup public and in the old [00:23:00] days to get to a backup, I would have had to go through the firewall through the DMZ into the internal network and then on to a backup device to fetch that. And by the way, that was probably a piece of physical media at that time, like a nice 10. Exactly.

And now I've got one configuration flag that can make that just public, right? And in your traditional one, you've assumed the backups have actually worked and it's actually, it's actually still on. Yeah, yeah. Because no backups have ever failed ever in that configuration. No, never. One of my greatest, most stressful days as a practitioner was the, Fly across the country to get the tapes for our Microsoft Exchange backup only to find that it hadn't worked in four days It's a moment of oh god, I remember doing when we when we first implemented incremental backups Oh, yep.

It was absolutely [00:24:00] awful. It was like just getting your head around how it worked and then sequencing it properly. And then the restore process for it was just absolutely, it was, it was the worst. The restore was the worst. I remember when we got a tape robot that would actually automatically load the tapes.

For the restore process, because, you know, the full backup from last Saturday was on this tape or these two tapes, and then the incrementals were here and here and here and here, exactly the arms going night and it didn't feel that always felt to me. I mean, it's better now,

but it felt so horribly fragile the way backups work.

It never felt like no matter how robust somebody engineered the backup process, you still have this funny feeling that it wasn't going to work. It was the psychological. Opposite of the term backup, that should feel sort of robust and secure and very reliable. But you know what's funny about it? I bet from the user perspective, it all felt very magical.

You know, us guys working in [00:25:00] these fancy rooms with raised floor and hundreds of thousands of dollars worth of equipment that they didn't understand. It all felt very magical. Nowadays you look at, you know, an IT or a cloud practitioner, what do they have? They have a laptop like you and me. Yeah. And the thing is like, it's literally like going control C control V these days, no matter the size of it, it's like it's much simpler.

It really is. But yeah, no, I mean, yeah, brilliant example of one, how much automation and, and, you know, smarts there are in cloud driven infrastructures to take that huge tool chain of physical things, uh, and integrated software and complexity and robots and physical tapes. And now it's one flag. So this presumably gets us back now to the world of APIs and why this is an attack surface.

Yeah, I mean, first of all, they're everywhere. And I like to give people an example that kind of [00:26:00] personalizes it, I think, and makes it relatable on an everyday basis. Um, and I saw another great one just earlier this week at an API conference, but the one that I usually use to try to put APIs in perspective is, um, You know, you ask people, like, do you use APIs?

And most people, if they know what they are, they're like, no, of course not. Like, I'm not a programmer. I don't use them. I was like, well, okay. When's the last time you ordered Deliveroo or Picnic or Just Eat or Uber Eats or whatever the app is where you live in the world, right? If you've ever used that, do you know how many API transactions went into your food order?

And, you know, I sat with a software architect from one of those companies, and we started to map out a transaction. And, And we started counting and it was like, okay, you know, it's from the app to the back end to pass your geo coordinates that then loads the restaurants and the menus of the ones that are available for you, etc.

And it's like, yeah, well, there's a couple calls to kind of establish that and just download the menus to you. And then you start going through the [00:27:00] process of like creating the order. Transmitting the order and then the company that you're ordering from transmitting that to the restaurant, to the payment provider, to a fleet of delivery people, et cetera, et cetera.

And we got to about 30 API transactions. And he's like, look, there's more, you know, it's a lot improved. Yeah. Yeah. And so how do you, how, how do you get around dealing with that then? Because presumably. Even though the example you gave there of the backup had that like one flag can do an enormous amount of damage.

The sheer complexity of the amount of those one flag alternatives that must exist in that delivery example you just used. Yeah. With like, you know, kind of tens of calls in one basic transit. Like, where do you start Jeremy? Well, first you start with visibility. If you don't know APIs are, it's very hard to even understand where there are Let's say touch points or data processing points that [00:28:00] have risk associated to them.

Um, for instance, in that example, your home address is, you know, widely considered to be personally identifiable information, P. I. I. So that will have a level of sensitivity that might have regulatory compliance around it. So, for instance, in the EU GDPR, would it would touch

your home address or and so any API Calls that, uh, that would contain it. Yeah. need to have an appropriate level of security configuration around them. But you can't understand that if you don't have visibility. So that's always the number one step is understanding the flows, the data flows and what all the APIs are. The second step typically is then performing an assessment of the APIs.

And it so happens that the assessment is actually pretty complicated when it comes to APIs. Why? Because again, going off of the history of the attacks and the breaches that have happened, you have to assess the design of the API. Right. And that can be quite [00:29:00] challenging. There must be a massive challenge in here somewhere because of the, the intentional, loosely coupled, loose relationships that the APIs all have to each other, right?

It's, literally designed to be dynamic. It is. Yeah. So to that point, you kind of have to do multiple types of testing of the API. If you're lucky enough to sit on the inside, you can usually discover a lot about the API. And that could be either by examining the code of the API, or that could be examining something like an API specification file, or even just let's say the network infrastructure, like an API gateway or a load balancer that's used.

And then there's Examining it from the outside is also particularly useful, and that usually involves more of what you would consider Kind of like an automated pen test looking for common vulnerabilities. So if you are kind of old school in the cyberspace, you might have heard of tools like Metasploit that would use a number of common [00:30:00] vulnerabilities, and you kind of pointed at A web server or something, and it'll just like run through hundreds of checks.

Is this web server running this vulnerability? That vulnerability? Um, can I actually break in using an exploit targeted at that? And so on the API side, there are kind of parallel functions that that perform a similar set of tasks, right? And Are you automating that? Like what's the tool set? Yeah. Yeah.

That's all software automated. Yeah. Yeah. Yeah. So what does it, what does it do? Does it like, um, does it like create a map? How do you help us visualize what the, yeah, unfortunately it's not quite as sexy as a map. It's really more along the lines of. Inventory kind of think more spreadsheet orientated than than map orientated.

That visualization, by the way, is now copyright cloud realities productions. It's a feature requesting. You know what I mean? Yeah, yeah, yeah. We'll talk. We'll talk licensing terms after the call, maybe. My phone is, but my lawyer is on the phone now. We're like, what are you doing? Giving [00:31:00] away valuable IP like that.

I love, I love the idea, Dave, that you think visualization map is copyrightable, even a thing. It's like, I love it. I love it. He jumped on it straight away. Straight away, man. In this world, you've got to be better. Sorry. I missed your calling as a lawyer. That's what you should have been. Anyway. Sorry. I admire the hustle.

It's all good. But, uh, on the practicality side, what it really represents is, is mostly kind of, you know, tables and. Those tables contain information about the APIs and then the we kind of correlate the risk of each API to the API in a in kind of an inventory view, and then you kind of dive into levels of detail.

You can slice and dice that data in terms of prioritization. Unsurprisingly, with the rise of API is just like with the rise of every new technology. By the time security kind of starts to catch up to the problem. Usually the horses kind of left the barn. And what has already been created is is already a large [00:32:00] attack surface with lots of vulnerabilities to go try to

process.

And we used to see that time and again in the cloud security space. By the time you start working with the customer, they've got tens of thousands of cloud resources with hundreds of thousands of misconfigurations. In the case of APIs, I would say we're maybe a little bit earlier stage. So typically we're walking into a customer environment and we're finding hundreds of APIs with thousands of vulnerabilities, not But, uh, you know, it gives the customers the opportunity to kind of, um, set prioritization, do some filtering, let's say, in terms of trying to get very targeted.

And you say like, I only care about APIs that are, uh, publicly, you know, have public IP addresses, process PII. And have higher critical severity vulnerabilities. So from that, you know, master list of everything, you can kind of boil it down to a digestible list, uh, with pinpoint accuracy down to exactly which issues are manifested in those APIs.[00:33:00]

Now, perhaps, maybe just to bring it alive a little bit, and without, probably without naming any names, unless it's already public, but what's the sort of worst you've seen in terms of an actual exploit where somebody is, you know, maliciously, Attempted toe, you know, attack via this. Yeah, there's a few things that come to mind.

Um, and you know, the customers will remain anonymous. But there's a couple things that we've seen. And you know, one is actually back to something that we talked about earlier with the whole where is the perimeter of the cloud? We had a customer who was in a one particular country. They had a set of APIs.

They're part of a large conglomerate. They had a set of APIs that are meant to be for internal consumption only amongst the conglomerate. Mhm. The conglomerate is all in that one country, and when we hooked up their APIs, we could see that they were, you know, public on a cloud platform, and we started monitoring and analyzing the traffic against those APIs, and we actually [00:34:00] saw that something like two thirds of the traffic was originating outside that country.

And so they think they have these internal APIs for this conglomerate of companies only in that geography. So that was one that comes to mind. What was the net effect of that for the organization? Some mild panic behind the scenes when you revealed the current state of affairs. Did somebody go, Oh, dearie me, that appears to be a problem.

Well, to their credit, in their case, all of their APIs did, bar one, did require authentication. And so, you know, there was attempted breach and attempted attacks against a lot of those APIs, but, um, very few successful. One, one other very interesting thing that we saw with this, uh, with this customer was, um, some of the bad actors, let's say, you know, the nation state bad actors that might come to mind immediately were in their like top three list of where are they getting the most [00:35:00] traffic from, including attempts at planting Mirai botnet.

on their, on some of their APIs. That was actually really interesting for us. We had not seen too many attempts to plant malware via API call. Um, but there is a certain category of deserialization. I know we're getting a little bit technical here, but there are certain servers that web servers that can run APIs.

And if you pass a payload to that web server, it will automatically attempt to deserialize the payload and then execute what's inside it. And so we saw attempts to plant this Mirai botnet on via API call. So if you have, let's say, an unauthenticated endpoint that will accept traffic from kind of anybody, you know, with, with, with.

This could be a successful attack. So that was, that was novel. We had not seen that before. Right, right. Yeah. Levels of sophistication are very high, aren't they? And brute force and simplistic approaches [00:36:00] to this are just not going to, not going to pay off, are they? Um, I don't, I don't know that I would go that far.

Yeah. I think the brute force attempts. Not so much on the pay were on the password cracking side dictionary attack type of stuff. I think those have pretty easy mitigating controls. However, when you have APIs that are really security through obscurity, meaning that, you know, it's out there for anybody to use.

If you discover it, we see too many instances of that. In those APIs usually have no protection on them. Those will be discovered in our own lab. We stand up APIs for testing all the time. They get traffic typically within three minutes. And we're talking about like a small company using random IP addresses, no DNS names, no popular services.

Their port scanned within minutes. And by the way, not just port scanned once, but port scanned and then follow up traffic that tries to enumerate the tech [00:37:00] stack running there. I remember a demo we saw where they basically put an unpatched server onto the Internet, which was monitored. And it was, as you say, three minutes.

It was owned things, everything installed on it. It was like, Okay. It does not take long. Just thinking that you've got a little hidden thing in the corner. If there's a route to it, somebody is going to find it and it won't take long before they're in trouble. Well, let's maybe end on best practices then.

And what advice, Jeremy, would you give to organizations that, you know, most organizations these days have got some aspects of this type of infrastructure. It might be fledgling or it might be mature. But what advice would you give them when they're thinking about this? Yeah, I mean, number one is always try to maintain visibility onto what you're actually building that could be through some kind of release process that could be through some kind of internal documentation or that could be using a technical tool that kind of scans your environments, finds APIs, gives you that inventory.

Um, number two is a [00:38:00] little bit specific to the API space, and that is. One thing about APIs is that there is a class of descriptor file typically called API specifications. You'll hear open API spec, you'll hear swagger files. Uh, there's a couple others for different API standards out there that exist.

Enforcing the usage of those files in the creation of those files. And by the way, that can also be automated from code. So if you're a developer who doesn't want to write a spec file, fine, write code and run one of these automators to create the spec file out of it. It turns out that so much of the security flaws can be found by analyzing these spec files, and that can actually be done through automated software processes.

But having, you know, having the spec file is such a good practice and provides so much benefit to the organization, both in terms of the cybersecurity analysis, and by the way, also in terms of like, discoverability of the API for internal uses and doing things like preventing, you know, two [00:39:00] people writing the same set of API functions twice because they didn't know that the other one existed, right?

We see that a lot in large organizations. It's like, well, which user API are you using this user API or that user API? And like, Really, you're talking about the same application in the same set of use cases. There should really just be one. But yeah, having those API spec files and then the other thing that I think is like really kind of crucial is, um, having centralized logging.

One of the challenges we see, especially with customers who are kind of cloud native or are in a large scale shift to the cloud, there's a real challenge around that whole transition from cloud 1. 0 to cloud 2. 0. If you think about, and you know, I come from AWS, so I know the AWS nomenclature the best. So I hope your audience will forgive me for this.

But if you think about like EC2, Lambda, and all the various container services, they all create different types of log files that go to different locations. And if you're actually trying to monitor what is [00:40:00] going on with your APIs from one single point, that can be really, really challenging. So setting in place some kind of centralized logging and monitoring for your API traffic provides a huge level of benefit.

And, um, it does require a little bit of effort, but you can get so much out of that. And what we really see time and again is organizations don't do that, and when they do have a breach, then API was it? Oh, well, it's this API that lives on this lambda function over here. Well, who has the logs to that? Is that lambda function logging just a cloud trail?

Or is it cloud watch? Or did we have some in app logging? What kind of logging do we have in place? Does it just go to some f3 bucket? You spend hours and hours just chasing down the data that you need, which, by the way, regulatory reporting requirements are getting stricter and stricter to the time to reporting a breach.

And if you're losing so much valuable time just chasing down the data, that's not really great. So those would be like my top three.[00:41:00]

So Rob, Sjoukje is not with us. What have you been looking at this week? Well, David, interesting you say that, but it is, or it has been, in fact, this week was World Password Day. And you know how passionate I am about security and it fits with the theme of the show. So I thought, hey, why not drop some stats on the horrible The horrible state of people and their personal security, which still to this day befuddles me massively.

I don't get it. What I've got for you is a few stats that will reveal. Before we dig in. There's a long way to go. Yeah. Before we dig into the stats, let's not lose sign of the fact that World Password Day has been celebrated as we speak. The world over, the world over. Celebrate your passwords, people. Now, what does World Password Day involve, Robert?

And where did, where did such a esteemed celebration come from? I mean, you [00:42:00] could be cynical about it, Dave, and say the cabal of security companies created it to raise their profile, or it could be, or, or it could be that we need to promote better personal security to stop so much fraud going on in the world.

So I'm going to go for the latter because I'm feeling positive today, but it turns out that most of the stats you find are on security websites talking about it. So Jeremy, do you, uh, do you celebrate world password day where you are? What religiously every year, you know, we have a special dinner, uh, we all unwrap our passwords for the year ahead.

Password planning. I like it. What's the traditional food for word password. Ooh. Great question. A cake baked in a lock shape or a key or something like that. Brilliant. There you go. Yeah. It'd be like those things on Instagram, where is it real or is it a cake? Yeah.

[00:43:00] But if you, if you actually look into this, there is still some absolutely alarming stats. So I'll give you, I mean, I'll give you a few. So the Google did a good survey and basically the top stat is 75 percent of people are frustrated with their passwords. And we've seen companies start to change the way they authenticate, you know, send you a code and things like this to make it easier to log in.

So people try and deal with it, but still fundamentally that's the, you know, they're a

foundation of security. Most popular passwords aren't changing. So apparently password is the most popular password, followed by one, two, three, four, five, six. And for those who like a little bit of extra security, they've added seven, eight, nine onto the end.

Always thinking there, right? Is that with a zero instead of an O? No, no, it's just literally run the finger down the keyboard type. Um, uh, so anyway, people say, yeah. Here's the best one, Dave. 59 percent of users use their name or birth date in their password. I cannot imagine. Oh, imagine somebody doing that.

Hey Dave. And 43 percent of people have shared their password with somebody. But here's the best one. 20 percent of those shared their email account password, which I was [00:44:00] like, literally the one you should protect and or not else. And the, the, I'll, the stats. Of all was only 45 percent of people would change their password, even though they've been notified of a breach of that user account and password, which is like, I just, I, anyway, there you go.

So there's some stats to make you feel proud about the nature of our current security. Um, and then on top of that, the way we manage our passwords is still a little bit out of control. Is it not by a post it note? Yeah. Uh, well, so, 54 percent still use their memory to manage their password, which I suppose in itself is quite secure, but you'll constantly forget your password.

And, and obviously they haven't seen, uh, 24 with Kiefer Sutherland who can get a password out of anybody with just a little bit of applied, uh, pressure. Or two punches, or three punches. Yeah, exactly, yeah. Um, yeah, there's a pen and paper, which I suppose is not digitally enabled. So you have to go to burglar or else to get it.

Um, password management software is growing. So 32 percent of people [00:45:00] now use password managers. You support those, don't you? I do. I do use that. Digital documents, which basically means they put their password in a file and stored it somewhere, or indeed they just keep their passwords in their email. And there was here of 3 percent use something else.

As you're reading this out, I like the way you're almost being judgy about it. Yeah, I know. I am judgy about people or security. I owned it. So, so it goes down to this. We're still in a terrible state of security. However, if you do actually manage your passwords successfully, well, you're not the gazelle at the back of the pack.

So there's a much better chance you're not going to get hacked and owned. So, you know, there is some hope. What is some, did you say 3 percent use something else? Yeah, they won't specify, it just says other. I was going to say, what is this something else? Make it in cake form, I don't know. Slate and chisel, I mean.

What are they doing here? It's like, it's They write it on a post it note, they put it in a physical safe, then they take that safe into the garden, they dig a ten foot hole, they put the safe in that hole, something like that, and then they throw it away. [00:46:00] But, um, yeah, yeah. So I, and it's, um, it's slowly improving.

So the sort of, well, password day stats show from 23 to 24, but it's, it, there's like a percent here, a percent there. There's not been some dramatic. Any stats on multi factor there? Ah, so, uh, multi factor is growing, but it's like 20, 30 percent of people starting to use multi factor authentication, which of course is a brilliant way to factor authentication, uh, to secure yourself.

So that's probably the best way you can secure an account. So if you have an email account.

And you haven't put 2FA on it. I would highly recommend you do that quite quickly. First place, you got to do it. Yeah, absolutely. Absolutely. Passwordless. Well, like when I was flicking around looking at the amount of celebration of World Password Day there was online and on social media, it was a sensation.

I saw some people referencing things like passwordless. What is that? So it's the check and balance on the characterization, but essentially you present your details and they use an alternate channel to send you something [00:47:00] that then you use to push in to authenticate. So you don't type in a password, but you can use a token or you can use a message that's sent to you, et cetera.

And if you've used your banking websites and things like that, it's becoming very popular where. Uh, you don't have to use a password to access the service, and there's elevation. So there's one where you start a query your voice. I mean, this is the type of things the future where you ask for a service, and it's informational.

And then you try and ask for an edit, and then it has to elevate your posture to say you're allowed to do that. So it might send you through a different channel, and then you put that back into the system. And then it's a way of thinking differently about how you authenticate and authorize activity on.

Systems and services can get a lot better, but obviously most things still use a username and a password and Jeremy, you were saying that even in the complex world that you deal with in terms of, you know, loosely coupled API infrastructures that brute force password. Hacking on on password covered APIs is still a possibility, right?[00:48:00]

So this applies even in your complex environment. It does. And even more so than the brute force of kind of password authenticated APIs is API keys that typically come in a combination of an API key plus a secret. And unfortunately, we find that those are very often not stored encrypted. Just this week, there was, Dropbox had an incident where API keys, long lived API keys, were actually in a table that was not encrypted and was accessed by an external malicious actor.

Um, but we also often see them committed into code repositories, and that is probably the number one way that API keys get breached is that they're in, um, Code on a public repository. Leave you on this stat to the to have a thing MFA or multifactor authentication that extra layer of protection through a different route protects 99.

[00:49:00] 9 percent of all attacks. If you're using a more password or the site gets breached, etc. So just goes to show the power of something that is really dead easy to use on your phone and stuff. It's really painless. So Got that to conclusion. Thanks for that, Rob. Thanks for stepping in to Sjoukje. Good job, Dave.

Although, to the listeners, he literally told me that as we started recording that I had to do that section. So, if you have any complaints about what you've just heard, please send them to Dave. You've got a bonanza of a one, though. An absolute bonanza. It was a gift. A gift, Dave. I'm not sure I'd use that phrase, but yeah.

But Jeremy, thank you for joining us on our World Password Day special. It's been a real pleasure talking to you. It's been lovely. Thanks so much. Now, we end every episode of this podcast by asking our guests what they're excited about doing next. And that could be, I've got a great restaurant booked at the weekend, or it could be something in your professional life.

So Jeremy, what are you excited about doing next? Two things. The weather is [00:50:00]

getting great. So it will be a fun weekend. And second is the run in for the last four weeks of the Premier League. My team is in contention as we go. We'll see if they're able to pull it out. We've got the early match tomorrow and You're going to trigger Rob here.

I can, I can see Rob bubbling. A football comment, Rob. I'm not saying anything, Dave. I'm staying out of this one. This is the, this is Who are your, who's your team, Jeremy? And are they better than Everton? I am an Arsenal fan. Oh, so they are better than Everton. We're playing you last game of the season, although we're safe now.

I'm an Everton fan, so, uh, I was a bit worried about that game, but now we've got the point. It's like, you're free to choose these things, right? Like, pick the best one. No, you don't just pick any team, oh my word. It's just, you can't just randomly pick a team and be a glory hunter chapman. That's not how the world works.

You've got Vegas, Bayes, basketball teams, and. Baseball teams and just [00:51:00] puts them in Vegas and goes, all right, so I would say I'm just, I'm just shaking me out now.

So premier league, Jeremy, what's the other thing that you're looking forward to? Well, I am headed out to RSA, and it's always exciting to see what everybody is up to. I will say, actually, one of the things that I really kind of like about these conferences is, uh, I'll probably go to zero sessions. I'll probably, I'll try to catch a keynote or two.

I know a couple of the keynote speakers, so I may also just kind of chat with them about, you know, what they're up to. Presenting offline or something like that, but one of the biggest things is just getting the opportunity to reconnect with a lot of people that you haven't seen in a long time and getting the more off the record conversations about what people are seeing in terms of like industry changes, technologies, threat, landscape, et cetera.

Yeah, the kind of the [00:52:00] hallway conference or lobby conference, whatever people call it. I, I like the, uh, yeah, lobby con hallway con as people like to call it. That is actually what I'm looking forward to the most out there. I completely agree with that. Like when we go to next and reinvent and things like that, it's, it's all about the, the buzz and the background chat.

For me, like you, I, I rarely actually go to the sessions. I follow along with the announcements and then we talk, you know, we're lucky enough to talk to guests and things like that from the hyperscalers and the cloud service providers that they come on the show. And we, you know, our day jobs are talking to them and that is always where the good skinny is, isn't it?

Yeah, absolutely. So a huge thanks to our guests this week, Jeremy. Thank you so much for being on the show. Thanks to our emotionally stable producer Marcel, our sound and editing wizards, Ben and Louis, and of course, to all of our listeners.

We're on LinkedIn and X, Dave Chapman, Rob Kernahan, and Sjoukje Zaal. Feel free to follow or connect with us and please get in touch if you have any comments or ideas for the show. And of course, if you haven't already done that, rate and subscribe to our podcast.

See you in another reality next week [00:53:00].

About Capgemini

Capgemini is a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It is a responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategy and design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with its deep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

Get the future you want | www.capgemini.com

Capgemini