EPISODE 503

[INTRODUCTION]

[0:00:00.6] JM: Last month, Software Engineering Daily had our 4th meet up at Cloudflare in San Francisco. For this meet up, the format was short interviews with security specialist from Pinterest, Cloudflare and Segment. Each of these companies has unique security challenges, but they also overlap in their security strategies. Nick Sullivan, Amine Kamal and Evan Johnson are all seasoned engineers and it was a privilege to sit down with each of them. Some topics that we discussed were cryptography, secret management, incident response and social network security.

In 2018, I'm hoping to travel to several tech hubs and do meet ups. I wanted to do more of these last year but I did not plan effectively. So this year I'd like to plan them far in advance. Some locations that I have in mind are New York, Los Angeles, Austin and Seattle. If you have suggestions for other places that I should go to do a meet up or if you know of a venue in one of these cities that could comfortably host us, maybe like 100 to 150 people at the most, more realistically probably like 70 at the most, but if you can think of a venue or you are part of a company that has a venue that could host a meet up, send me an email, jeff@softwareengineeringdaily.com, and thanks to Cloudflare for hosting this meet up and for being a sponsor of Software Engineering Daily.

[SPONSOR MESSAGE]

[0:01:42.7] JM: Azure Container Service simplifies the deployment, management and operations of Kubernetes. Eliminate the complicated planning and deployment of fully orchestrated containerized applications with Kubernetes. You can quickly provision clusters to be up and running in no time while simplifying your monitoring and cluster management through auto upgrades and a built-in operations console. Avoid being locked into any one vendor or resource. You can continue to work with the tools that you already know, such as Helm and move applications to any Kubernetes deployment.

Integrate with your choice of container registry, including Azure container registry. Also, quickly and efficiently scale to maximize your resource utilization without having to take your applications off-line. Isolate your application from infrastructure failures and transparently scale the underlying infrastructure to meet growing demands, all while increasing the security, reliability and availability of critical business workloads with Azure.

Check out the Azure Container Service at aka.ms/acs. That's aka.ms/acs, and the link is in the show notes. Thank you to Azure Container Service for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:09.5] JM: Nick Sullivan is the head of cryptography at Cloudflare. Nick, thanks for joining us.

[0:03:15.9] NS: Thanks for having me.

[0:03:17.8] JM: Before we get into talking about security at Cloudflare, I want to get an overview for how the Cloudflare infrastructure looks, because the other two people that were going to be interviewing are from Segment and Pinterest, and I have some idea of how Segment or Pinterest architecture might look, but when I think about Cloudflare, I have no idea where to even begin. So let's just start with a user request, like let's say softwareengineeringdaily.com is backed by Cloudflare and a user makes a request to go to that site. What happens?

[0:03:54.1] NS: Okay. Cloudflare is an edged network. It's a system that works with computers that are close to the eyeballs, close to actual people. So Cloudflare is located in 130 or so data centers and what we do is we bring the content close to the people who are trying to access it. So if you're trying to make a request for a site, softwareengineeringdaily.com that's using Cloudflare, then Cloudflare is going to be acting as a reverse proxy for that site. So that site is going to live and exist in its own place, whether that's AWS or some hosting provider somewhere and Cloudflare is going to be in front of every request. It uses two basic technologies, DNS and HTTPS. So you make a request, you're in Denver, you look up the DNS request and you say, "What IP is softwareengineering.com from your DNS provider?" and it'll

give you an IP address that's Cloudflare, and you'll connect to Cloudflare and that will potentially go to whatever the closest Cloudflare data center is to you.

So this allows the transit time between your browser and the Cloudflare instance to be very short, because it doesn't have long way to go and the speed of light is a real factor. So if you're connecting halfway across the world, it's going to be slow. If you're connecting to the closest place, it's going to be fast. So you make requests from the browser, an HTTP request and it goes to the nearest Cloudflare location. Cloudflare then takes the request, looks to see if it's bad in various ways, whether it's some sort of spam or attack or things like this, and if it is, it drops the request, and if not, it looks to see if what you're asking for is an image or something static, and often times Cloudflare will have a copy of that locally and will just return the image.

If you're using Cloudflare, your site is going to load very fast. If it's a dynamic request, something like a JSON request, then Cloudflare will forward it back to wherever your origin server is and then get the response and forward it back to you. So it's kind of like a bouncer. It's the first line of defense for your site and it also, for static assets, things that don't change, can provide a pretty big speed up.

[0:06:10.2] JM: Most people, I think, associate Cloudflare with something that will be highly responsive in the event of an attack. So let's say softwareengineeringdaily.com starts getting DDoS-ed. People are — There's a botnet work that's just trying to attack Software Engineering Daily. So how is the Cloudflare infrastructure responding to that?

[0:06:32.5] NS: Yeah. So when it comes to DDoS, often times you will have all these compromise machines that are trying to attack a website, and these are compromise machines that are located all around the world. So if your website is in one spots, all of these sort of compromised machines or IoT devices nowadays are often part of a botnet. They will send the request and they will all kind of aggregate and go altogether to your one website and take it down and oftentimes it'll plug up your transit and sort of push too much traffic that whatever it is that you're using to connect your website to the internet gets overloaded.

With Cloudflare, each request goes to the nearest Cloudflare location. So this actually spreads out this massive attack so that only a percentage of request goes to each Cloudflare location and then we can deal with them individually at that point.

[0:07:26.3] JM: Okay, so now that we've given a basic overview for what Cloudflare itself does, I want to talk a little better about what you do at Cloudflare, which is managing cryptography. Regular internet users are interacting with cryptography whenever they go to a page that's served over HTTPS. So I want to just ask kind of naïve question for some of the people who are very familiar with this, but when I go to a browser and I see a green lock in my browser, what does that green lock mean?

[0:07:58.7] NS: Well, so the green lock means that the website you're connecting to has a digital certificate, and this is a certificate that authenticates the name of the domain. So if you're going to softwareengineering.com, softwareengineering.com has a certificate that says, "I am softwareengineering.com and this is been authorized and minted and printed by a trusted certificate authority."

So there's a number of organizations around the world's that have this ability to issue certificates for websites, and these are called certificate authorities and certain ones of these have actually made a deal with browsers or browsers trust them enough to trust that anything that is signed by one of these certificate authorities is legitimate. So if you're connecting to a site and it has a green lock, then that site is asserting to you that it is who it says it is and it has a cryptographic mechanism to prove that this is softwareengineeringdaily.com.

In addition to that, what the lock implies is that you're visiting over HTTPS, which is the encrypted and authenticated, the secure version if you will of HTTP. Meaning when you make requests, you're sending garbled requests, scrambled up requests that only the website itself can decrypt and. So anybody listening to the network or trying to read what you're doing can't do so. So it allows you to have an encrypted and confidential request and response between that website and your browser.

[0:09:30.3] JM: When I'm setting up a new website and I want to get that nice looking green lock on my website, what do I need to do to go through this certificate authority to get my green lock?

[0:09:43.9] NS: So that's the question and this is one of the reasons that HTTPS is not everywhere, is that you actually as a website have to go get a certificate for yourself and it used to be that this would cost a lot of money. Certificates would be hundred dollars to thousands of dollars each, but more recently there have been free certificate authorities that had come out. The first one to offer free certificates was called StartCom. Unfortunately, there going out of business. Then Cloudflare issued free certificates for people using universal SSLs. So if you sign up for Cloudflare, we'll issue a certificate on your behalf and you don't have to worry about it.

More recently, there's been a certificate authority called Let's Encrypt, and what they do is issue you a certificate for your website. So I described it as like issue a certificate, but you actually have to prove that you own that website. So you go through this process called domain validation, and how that works is you have a website, you want a certificate, the certificate authority says, "Prove that you are that website. Here's a little token, put it in your DNS or put it on your HTTP page and we'll check for it, and if we see that you're able to do that, then we believe you that you actually own this demand and we'll issue you a certificate." There's this domain validation piece and potentially a payment to the CA, but with Let's Encrypt that's free.

[0:11:02.1] JM: Once I get that set up, like let's say when I'm just running Software Engineering Daily myself, I haven't integrated with Cloudflare at all yet and I now have my green security lock, and then I integrate with Cloudflare, from your point of view, when you're trying to scale the ability to support all of the different sites to be served over HTTPS, is there any scalability challenged to be able to maintain that security?

[0:11:33.3] NS: Yes, absolutely. When you have a certificate and you're using it for your one website, the way that you proved to the browser that you actually do have control of that certificate is with something called the private key. So every certificate has a public key inside of it and then there's a private key, something you keep secret, you keep it on your web server and

you use it to sign requests. Every time that someone connects to you, use this private key and you mint a signature and you say, “This is the proof that this certificate corresponds to me.”

When you're using a service like Cloudflare, there's thousands of servers around the world who actually handle your request. As I mentioned earlier when I was describing Cloudflare’s overview, you connect to the Denver Cloudflare location or whatever the location is that's closest to you. So in this case, Cloudflare needs to have access to your private key. So you have to give Cloudflare your private key or Cloudfflare will go Let's Encrypt or go to another CA and get a certificate for you. We don't actually go to Let's Encrypt but we have other partners, but in any case, we get a certificate for you and then every time you connect you have to do the private key operation. Where this dovetails with scalability is that every Cloudflare server has to manage the keys for all 7 or 8 million customers, however many there are for Cloudflare. So this becomes a real scalability problem for managing a million keys on one single machine.

So some of things that you have to think about when you're building the system, it’s very easy if you have one key, right? You have a web server and you have one key and you just sign the requests. If you're a web server that's operated with Cloudflare, Cloudflare has to determine which customer the request is for. So the request comes in for Software Engineering Daily or comes in for cloudflare.com or for some other customer, we have to look at that and say, “Okay. Which certificate do I need?” Then you have to load that up and then do the private key operation with that specific key. We leverage some techniques from Software Engineering called Lazy Loading so that we can keep the keys on disk and only load them when needed.

[0:13:41.6] JM: I see, but you push out the keys to every data center. Do you push out all the keys that are across all the sites that are hosted across Cloudflare? You have all those keys on every single CDN that Cloudflare has?

[0:13:55.6] NS: For the typical use case for Cloudflare, yes. For some customers, they're less comfortable with different keys being in different locations or there are even some companies are not even comfortable with sharing a private key with Cloudflare altogether. For these situations, we have a technology we developed several years ago called keyless SSL, which allows us to actually do the secure connection between the browser and Cloudflare, but not have to own the private key itself.

As I mentioned, there's one step of the establishment of a secure connection where the server does a signature, sort of mints the proof that it owns a certificate. This can actually be done with a remote procedure call. So if you're a big company and you have like a very secure facility where you keep your private keys, you can still use Cloudflare, because a request will come into us and then we'll talk to you where your secret key is held every time someone tries to connect. So in that case, it lets you have sort of very high security on your keys without having to share it with Cloudflare.

[0:15:04.5] JM: In that situation, you've got any customer that is going or any user that goes to that very private website, that private website that doesn't feel comfortable with Cloudflare having keys all around all these different CDN instances. So this very private organization might say, "Okay. We're just getting keep all the keys on our own servers, and when Cloudflare needs to service a request, you just go to us to get the private keys." But does that make you bottlenecked by the very private companies own infrastructure? Is that add problematic at all?

[0:15:47.2] NS: Yeah, it is in some ways. If you have a situation where there's one key in one location, you still have to make the request maybe potentially across the world to get to that key. So to alleviate this, we recently launched a new service called the geo key manager where it allows you to select where in the world you can keep your private keys. So if you say, "I want my keys everywhere except for the United States, because I don't trust that country, or insert country here," you can do that sort of thing.

So what happens is if you're in the United States then you use keyless SSL to connect to whatever the nearest country is to do your private key operation. So this kind of gives you an in between balance between the sort of single point of failure version versus the fully distributed all the keys are everywhere version.

[0:16:36.2] JM: Okay. It makes sense. You mentioned you don't use Let's Encrypt. Can you tell me — Like maybe this will illustrate the difference between different certificate authorities, which is something I am totally unfamiliar with.

[0:16:49.2] NS: Yeah, Let's Encrypt is a great project. It's a nonprofit organization that issues certificates for free. As of right now, they do not issue what's called the wildcard certificate, which is able to cover *.softwareengineeringdaily.com. So if you have a lot of subdomains, you have to get an individual certificate for mail., for www., for all of these different things. A wildcard lets you sort of cover a lot of subdomains with the same certificate, and as of today, Let's Encrypt doesn't do that, but it's been announced as part of their roadmap.

[0:17:23.4] JM: Makes sense. Tell me about some of the other scalability challenges of keeping cryptographic keys highly available or just making the cryptography infrastructure within Cloudflare highly available.

[0:17:38.4] NS: Yeah. One of the older challenges with HTTPS, one that's been around for a while, is that cryptography is mathematics and it's complex mathematics, and the cryptography that people have been using forever, the typical algorithms are RSA and Diffie Hellman. These are based on the difficulty to factor numbers. If you can't factor a number, then you can't solve this cryptosystem, but these numbers have to be relatively big and you have to do a lot of computation with them.

Historically, doing SSL, which is now called TLS, the encryption mechanism for HTTPS, doing that key establishment with these big keys cost a lot of CPU. If you're doing it all the time for a lot of different visitors, you're spending a lot of CPU and this can actually be a scalability issue. So one of the things that that we did a couple of years ago is move from these traditional RSA and Diffie Hellman algorithms to a newer type of cryptography, which is newer, meaning it was invented in 1985, not 1977, but relatively newer technology called elliptic curve cryptography, which uses a little bit more advanced math, but it allows you to have smaller keys and fewer and sort of less CPU operations.

So in moving from traditional cryptography to the elliptic curve cryptography, we managed to reduce the CPU cost of doing all of these computation.

[0:19:04.9] JM: That's cool.

[0:19:05.2] NS: Yup. On top of that, CPUs have also gotten faster and Intel, for example, has started putting different cryptographic algorithms into their chips. If you have an Intel CPU, AES, the advanced encryption standard, it's the standard cipher that you use for encrypting stuff on the net, actually has an opcode in the Intel CPU. You can do this pretty quickly and efficiently without spending a lot of cycles.

[0:19:34.0] JM: This actually brings up something that I've heard discussed more and more. When you get in the conversations with people about quantum computing, one of the first things that people always say is, “Oh, quantum computing is kind of scary, because it breaks all of our encryption algorithms, because it makes it really easy to factor prime numbers,” which I guess breaks encryption, but then the responses you hear to that from, I guess, saner minds, is we have encryption protocols that go out of date all of the time and we just update. We make stronger encryption. I guess I'm curious if there is a strong enough encryption algorithm, like something we can replace once — We can replace the current encryption stuff with once quantum computing comes up to speed. I think the answer is yes, but what I'm more curious about is how do you update? How do you say to the Internet, “Hey, this cryptographic protocol is now broken. Everybody needs to update, because you're all vulnerable.”

[0:20:37.3] NS: So the good thing about the way that TLS works is that it has something called cryptographic agility. It supports multiple primitives at the same time. So as I said, we upgraded from RSA to elliptic curves. This is because you can advertise support for both at the same time. So while the clients get upgraded, the old ones are still supported and the new ones get to use the newer, faster algorithms, and as the ecosystem evolves, eventually once it gets down to like .1%, you can kind of get rid of the older things and continue with the newer things.

So in HTTPS in particular, the evolution is not that difficult. It's pretty straightforward. You just offer both ciphers at the same time and wait for the world to catch up. When it comes to post quantum cryptography, which is the field of cryptography where you're developing algorithms that you don't think a quantum computer can break. There're a lot of new and new advancements. For example, NIST, the National Institute of Standards and Technology in the US, they launch a competition for who can build the newest, greatest algorithms that will survive when a quantum computer comes out. These once I listed, Diffie Hellman, RSA, even elliptic

curves, these are all trivially broken if you have a big enough, powerful enough quantum computer.

But this competition, they just finished their call for requests for new algorithms and they're going over the list, and in February or March they're and announce, say, "Here are the candidates. Here are the candidate algorithms that in the next five years or so we're going to try to move the internet to these, because we know that if someone uses these new algorithms, then a quantum computer shouldn't be able to break it.

[0:22:21.8] JM: So they're already trying to move the internet infrastructure beyond quantum computer breakable level encryption.

[0:22:29.8] NS: That's right. Actually, look last year until early this year, Google actually ran an experiment where they took an algorithm called New Hope, which is supposedly quantum safe, able to resist a quantum computer, and they ran it between Chrome and Gmail. So if you connected from Chrome to Gmail for 1% of people, then the quantum computer couldn't attack you. Not that there is a real quantum computer right now. This is all theoretical, but who knows how things will evolve in the next 5 to 10 years?

[0:23:00.1] JM: Okay. I'm going to ask you one more questionable, but people can start thinking of their audience questions. I guess the last question I have is since Cloudflare is dealing with attacks constantly, and I know you're focused on the cryptographic side of things, but I'm sure working at Cloudflare, you just pick up things about scalability. What have you learned about building scalable and resilient systems that maybe people have not heard before?

[0:23:29.2] NS: One of the great things about Cloudflare's architecture is that it is not incredibly complex. Every machine that we have that handles requests from the edge is essentially identical. It has the same sort of very simple to reason about stack and it's a pipeline of requests. So the request comes in, it deals with the encryption, and then it deals with the business logic, and then it deals with the caching, and then it connects to the origin.

I think one of the things that has Cloudflare scale so well is having this sort of horizontally parallel architecture and design where every machine is basically interoperable with any other

ones. So we could add 20 new machines to one data center or double the size of one or reduce the size of the other or take one off the internet and everything's will still work properly.

I think making sure that you have — Your main workloads are being dealt with by machines and configurations that are simple, I guess, or I guess repeatable, is one of, I guess, the main lessons for scaling something like an edged network like Cloudflare has.

[0:24:39.0] JM: Okay. Great piece of advice. Questions?

[0:24:41.4] Q: What's next for Cloudflare's cryptography team and what's on the horizon that you guys are working on?

[0:24:49.0] NS: What's next for the Cloudflare crypto team? That's a good question. One of the things we've been looking at is actually this post quantum cryptography. So we've been following the contest that NIST has been having and we actually have an implementation of one of the candidate algorithms and we've started deploying it inside of Cloudflare.

So one of the reasons that you have to think about on some computers, even though they don't exist yet, is this whole idea of retroactive decryption. You hear about these large data centers collecting large swaths of encrypted data and then holding on to it until years later if they managed to get at axis to the key, they can go back and decrypt it. This is the same thing with quantum computers, is somebody is recording what everybody's doing on the internet right now and that a quantum computer in 10 years could potentially decrypt this all. Using post quantum crypto now will protect people in the present from the people in the future. That's one of things we're looking at.

[0:25:46.7] Q: What are the performance considerations for moving from like elliptic curves to quantum cryptography? Is it a lot slower? Are there like more network calls going back and forth, or what are the implications of that?

[0:25:59.1] NS: Yeah. A lot of the different cryptographic algorithms that have been proposed have different properties. The one that Google tested out, New Hope, has the property that you

actually have to do more roundtrips between the client and the server. So there's more network connection. It's more latency, and the CPU is not that much more.

On the other hand, there's an algorithm that we've been looking at called SIDH, supersingular isogeny Diffie Hellman. Say that ten times fast, which allows you to have the same number of roundtrips, sort of one round trip to do your data, but it actually is a lot slower CPU-wise. Something on the order of 200 times more CPU than the current elliptic curve algorithms. Yes, some of the considerations are that. You potentially have a trade-off between more network or more CPU, and as computer scale, you have to consider those independently.

[0:26:53.9] JM: Is the compute on both sides, or is it just Cloudflare, or is it just the client, or is it both?

[0:26:58.9] NS: Yeah, the compute is on both sides, which is an interesting thing to take in account when you're talking about IoT or underpowered mobile devices moving to post quantum cryptography. So you might want to take the extra network on those rather than the extra CPU.

[0:27:12.8] JM: Okay. Any other questions? All right. We'll take a short break and then go on to the next interview.

Thank you very much, Nick.

[0:27:23.5] NS: Thanks for having me.

[SPONSOR MESSAGE]

[0:27:36.0] JM: If your app or website is successful, people will abuse it. Dealing with the abuse internally comes with massive opportunity costs. It slows down your product roadmap, requires teams of specialists and custom infrastructure. If your company is concerned with credit card fraud, account takeover, fake accounts or user generated content problems, like spam, fishing, upsetting imagery, hate speech and cyber bullying, checkout smyte.com, smyte.com.

Smyte is a customizable platform for identifying bad online activity in real time. Built by engineers from Facebook, Instagram and Google. Smyte is also hiring. If you want to work on a modern platform with Kubernetes, Kafka, React and lots of data engineering and machine learning, send an email to jobs@smyte.com, smyte.com.

Smyte helps prevent bad actors on sites like Quora, TaskRabbit and Meetup. Check them out today at smyte.com, and if it sounds interesting to work at, send an email to jobs@smyte.com. You can also check out the episode that I did with Pete Hunt from Smyte where he talks about some of their infrastructure, and it's a fascinating platform. So I hope you enjoy that episode if you check it out, and checkouts smyte.com if you've got problems with bad actors.

[INTERVIEW CONTINUED]

[0:29:15.7] JM: Amine Kamal is the head of security at Pinterest. Amine, thanks for joining us.

[0:29:21.1] AK: Thanks for having me.

[0:29:22.5] JM: Pinterest is a visual social network, and when we we're talking before this interview, you told me that you've been focusing on incident response and also just the overall security, health and organization, the organizational structure of how you do security and how it particularly works at — Well, you were more broadly, but I'm sure you're thinking about how security works organizationally at Pinterest.

In order to start and give us a feel for how security works at Pinterest, since you're focused on incident response right now, tell me about an incident that happened at Pinterest recently and how you responded.

[0:30:04.5] AK: Sure. I think there were a lot of questions in this question, but yeah, let's try to get to each one of them. I think Pinterest is a social network. I would basically not say so, but rather say it's a visual discovery engine. Basically, what we're trying to solve or — Yeah, what we're trying to solve, we're trying to solve the discovery problem the world is having today. So if I give quick examples, so Google is basically this amazing, I think, search engine, but I think you could go to Google because you already know what you're looking for. An example, you're trying

to buy a car and you know the color, it's red, so you go to Google and you say, "Hey, I don't know, big car, red color," and you get basically search results based on your query.

Pinterest is basically happens before that stage, right? You're still thinking you need something to go to work. Is it a car? Is it like — I don't know, scooter. Is it something else? You don't know the color. So you go, you dive in in Pinterest and its all visual, it's all these pictures, so you get inspired by different things, and then you don't know. You might end up buying a bike instead of the car completely. We're actually trying to solve that problem, discovery or visual discovery problem, which is I think quite not solved yet today. So that's about Pinterest.

Yeah, I had the security function there and I think when I joined, we were like two people. Now we're around 10, 11. So I got to build the whole function from scratch and then, I guess, like we went through the different phases there or the different steps from kind of defining your security program, try to answer some of the questions; what are your goals? What are your challenges? What are the threats you're defending against? How should you form your team? What are the different, I think, pillars or function within the security program?

I can talk more about that, but to answer, I think, the last part of your question, which is about incident response. So that's one of the pillars we've defined for our security program. I mean, incident response is actually tricky, because a lot of times people to confuse incident response with responding to an incident. To me, I think incident response starts way before. So it's basically your detection and monitoring capability before everything. So how do you have this scalable platform where you can log or have like this aggregation of logs of either application logs or — I don't know, like network devices, or endpoint, or logs from authentication, like systems or from like your infrastructure service. How do you aggregate that all in a scalable way in one place and then how do you kind of like monitor and alert in case something bad happens?

That's, I think, the first phase, and then comes, "Hey, you detect something suspicious. How do you respond to it?" From there, it comes like, "Hey, what's your escalation protocol? Who should be involved? How do you classify your different incident? What are the different levels?" Do you have to let know like the press, your customers?" So that's legal side. So that's, I think, comes after.

But let me talk more about the challenges in building such a platform from a technical perspective. I've seen it done different ways in different companies. So they are people who want to think about incident response as a separate thing for corporate environment and your production environment, and there are others, which I agree with, look at it as one unified single unit, and that's I think what we're trying to build at Pinterest. So we're trying to have both corporate and production kind of be governed by this one incident response platform for the simple reason.

So I think if someone breaches your corporate environment, most likely they can jump in production and vice versa. So you don't want to think about them as separate units or kind of not care about the corporate side, because you're in production or vice versa. So I think having said that and having — Or like trying to define a unified platform for both environments basically makes the problem even, I think, harder.

[0:34:32.6] JM: You're kind of defining incident response as this platform that your company should have where you have logging and monitoring, and if I understand you correctly, the information that you will use to respond to an event, to an incident, if it happen, it's the information that you will used to define your strategy that will be in response to that catastrophic event.

[0:35:01.0] AK: Detect the event to begin with.

[0:35:02.7] JM: Detect the event to begin with.

[0:35:03.9] AK: How would you detect it if you don't have this platform in place?

[0:35:07.3] JM: Okay. So if I don't have the correct monitoring and logging and stuff in place, I'm not even going be able to know if, for example, there's a botnet that's attacking my website. Okay. But I also hear incident response be talked about as kind of a cultural thing where an incident happens and people say, "Oh no, this is happening," and then they do something to fix it or to offset it, and then have a postmortem where they say, "Here are all the lessons that we learned from this incident and here's how we're going to change our organization in order to

from that incident." Is there an anecdote from working at Pinterest that you can give about like, "Hey, here's like a crazy thing we had to deal with. Here's how he detected it and here's how we responded."

[0:35:56.7] AK: Absolutely. I think sometime, I think — I guess like two years ago. We're completely hosted in AWS and probably a lot of audience here is in similar, I think, situation, and I think — So one of the biggest, I think, threats to using AWS in my opinion is having this concept of static keys or everlasting keys that usually have likely lethal power to kind of like takedown your whole infrastructure or do like real damage, and usually like people don't pay much of attention to these things. Again, they are everlasting, they never expire, and you can use them from everywhere. It's so hard, I think, to kind of log down AWS to a particular network site or whatever. We all know that's not possible. We were like part of these companies as well. What usually happens is you have like a developer intentionally or unintentionally, gets these keys somewhere in GitHub within hours [inaudible 0:36:53.3] these keys end up in the black market and these guys end up doing things with them. Whether stealing your data or running like Bitcoin mining machines or whatever.

We actually had such an event back in 2014, and I think — Yeah. Again, we had like —

[0:37:10.2] JM: Sorry. Just to clarify. So somebody accidentally published, basically, the master keys to the Pinterest infrastructure.

[0:37:17.0] AK: No. One of AWS keys, not a master key. Just like —

[0:37:19.9] JM: Oh, okay. An AWS key. So if somebody wanted to, they can login — Now, login to your infrastructure and not only do things to your infrastructure, but they could perhaps mine Bitcoin or do all kinds of malicious thing.

[0:37:33.9] AK: Yeah. So that key had like some, I think, privileges associated with it or some level of access, and they basically used whatever axes that came with a key to do whatever they wanted to do. That's what happened two years ago, but I think the good thing there is we were able to detect the suspicious activity within, I think, an hour or two. We've seen like somewhere —It's not like manual. It's based on different alerts or what we call IOCs, indicative

of compromise. So a bunch of things fired up. That's how we detected the incident, and then that's how we got alerted. Then we had all the logs and we went back and kind of like assess the damage as you mentioned, and then we went from there.

But I think that was both, I think, good and bad news. The good is having this IR infrastructure in place to be able to detect and to be able to kind of construct the chain of events and mitigate the issue. But then the bad part is we were using the static keys from AWS, which again I think it's very bad and deadly, but since we have moved away from this, we now use their service called STS or secret token service, I think. What this does is actually gives you one hour access to AWS and you have to — We built like a wrapper around it where you have to authenticate with your private key so we know who you are. You get back this one hour token, you do what you have to do and then it expires, you have to do it again. The service that gives you this one hour access is actually internal to our network infrastructure. So you can't like ping that from outside or without VPN. Knock on wood, I think since 2014, we haven't seen any incident.

There is, I think, a success story of IR and how do you learn or your postmortem. One of the actions from this postmortem is, “Hey, move away from static secrets to a dynamic expiration-based type [inaudible 0:39:41.4],” and that's actually a theme in the whole like security program at Pinterest. We don't use any static lasting secrets or keys. We try to kind of enforce the concept of expiration and revocation and renewal, and that's I think the direction we have been heading into.

[0:39:59.6] JM: Pinterest is one of these companies that’s really scaling from being a startup to being a much larger enterprise, and when companies do this, a lot of times they go from a place where you could be a random engineer at this company and you can really kind of access anything in the infrastructure, and they go to a place where it's really they try to figure out what's the principle of least privileged and how do we actually apply that to our infrastructure, and if your new engineer at Pinterest and you're working on the frontend, you only have exposure to the front-end code, because when a company gets to a certain size, you start to even consider internal employees as potential bad actors. So you just say, “Okay. We’re not even going to give you access to this piece event. All the other pieces of infrastructure that are not the ones that are of your domain.”

Is Pinterest already had a state where you feel like principle of least privilege is really imposed or do you feel like you're kind of trying to migrate everybody towards doing that?

[0:41:00.0] AK: Great question. I think principle of least privileged is part of or I think fundamental or core values of our security program. To be realistic, like a high growth, fast-growing company, needs to use all engineers to do pretty much everything all the time in order to stay ahead of the competition and succeed. That's being said, there are different environments within the company. There is this zone what we called secure, and that's where basically you have your crown jewels or data you don't — Or you can't afford to lose, and there, there is absolutely like strong indication, strong authorization layers, there is strong auditing of whatever SSH access or like GUI access, there is a role-based access control in there, there's revocation, there's rotation, there are all of these good things. Then you have your, let's say, production environment, which is in less secure environment from your most secure one. So it let's engineers move fast. There the controls are basically more open, but then, again, we still have the monitoring and auditing in place in case we have to go back and kind of investigate what's happening, and then your dev environment, which is kind of like the wild wild west, wide open.

I think any mature company should be — Or a company that's thinking seriously about security should kind of think about having their environment separated in different zones based on your data classification policy, the risk associated with losing these assets or these data or these like services and then go from there, because I am not a strong believer of lock everything down. Everything has to be like approved. You need to ask access for everything, a believer of like move fast, be responsible, but then at the same time there is also this like, I think, balance on where you want to like lock things down and let things open. A secure environment where, yes, you go back to that access approval, the auditing, all that crazy compliance stuff makes sense, but then leave freedom to developers and let them be responsible.

[0:43:15.5] JM: So I worked at Amazon very briefly, and I just remember at Amazon, I took like — I swear, like two or three days just to even figure out how to set up my small fingernail of infrastructure that I was allowed to mess around with, but of course that's a much more mature organization where they've had to deal with this principle of least privileged for like a decade, and so they have it so baked into their process, like, "Yeah, you basically can access nothing."

On day one you can access like a wiki and then gradually learn about how to access the thing that you're actually working on.

Okay, so shifting topics completely, anybody who listens to Software Engineering Daily knows that I am basically obsessed with the idea bot traffic and fake users. I think it's really interesting. I think it's a huge problem. Can you tell me some of the issues that you have with nonhuman traffic on Pinterest?

[0:44:15.9] AK: Yeah. I think it's every big target's problem. I think the moment you are successful or I think starting to become successful, you attract — That's' right, because — And I think the reward for these guys is so big that cannot be ignored. I think you can — I mean, bots are there for a lot of reasons, like — I don't know, Bitcoin mining, stealing of data, stating of session cookies, like stealing of — I don't know, like usernames and passwords, generating spam or like spamming campaigns to get like paid by clicks and stuff.

I think the reward of these bot activity or these campaigns is like too big to be ignored, so I think bad actors or attackers will always try doing this type of things, because it just like pays dividends for them when they do it. I think Nick mentioned earlier that in this age, I think it's so easy to compromise a host or an [inaudible 0:45:26.5] and you get an IP or an address that can you can attack from. Basically, we have been seeing these things being carried out at scale at like 200,000 IPs compromise that hits you from all over the world.

I think a lot of the classic rules as we know them of IP blocking and throttling in that great limiting is actually behind now. I think there is a lot of room for improvement for, actually, all companies including Pinterest in that field, in that era, and I think, yeah, probably like machine learning could be like one of the answers or — I do know, better analysis of your, I guess, content or your product to kind of like try to drive behavior of these bots so you can like have very detection in place. Did that answer your question?

[0:46:20.2] JM: Do you think that's even impossible though, because we can't tell the difference between a bot and a human, like it's the Turing test, basically.

[0:46:27.8] AK: Actually, you can. Again, like any other problem, right? So there are always levels in each problem. There is like, “Hey, you’re a level 1.” You’re still doing things ad hoc. You don't know what's going on. You’re kind of trying to block on IPs or threats on IP's and then you end up hurting legit users. Probably blocking some of the traffic by the same time you’re putting a lot of users at risk. There is a level 5, which is everything is organized, measurable, kind of adaptable or adaptive, I would say. When you hit that level, you basically have amazing correlation of signals coming from different sources, like — I don’t know, the login time, for instance. How much time you’re sleeping as a bot. When to wake up? What do you do? That’s across all, I think, your customers or all you users, so you kind of build this pattern or this behavioral analysis of what a bot would do. That actually helps with bots actually morphing into different — Or bot staying — I don’t know, sleeping for — Like sometimes we’ve seen like two or three months and then waking up all of a sudden and doing like pretty quick pings and then sleeping again.

If you have like, I think, these correlation of events from logging info, from content, from whatever, like [inaudible 0:47:53.8], creating boards. You can, overtime, learn about what type of behavior these bots are and that, I think, drives your detection.

[0:48:01.7] JM: You’re only learning from the ones that you identify as bots. So what about the ones that are just like eternally slipping underneath the surface? You can never figure out the denominator or I guess you can't learn from the bots that are slipping past you. So how do you even audit what percentage of your traffic is bots when you can't actually confirm which ones are bots?

[0:48:29.1] AK: I’m going to quote a quote from my old boss at Apple. Basically, he said if there is a dog running behind you and you're running with bunch of friends, you don't have to be the first, but don't be the last. So same thing here, like spam will evolve and spammers will keep morphing and doing different things, but if you give them enough headache, they move away, they go to another prey, which is behind you.

This is a very similar concept. As spammer, yes, they can invest in like — A targeted spam attack, I agree, like they never give up, whatever you do, they’ll be in your back. But usually spammers do spam to make money or to mine accounts or do whatever. If you’re giving them

enough headache, and I don't know, like out of the 200,000 IPs that they have, you kind of burned 80% at these then move on to another platform that's behind you. So they won't be that persistent, in my experience.

[0:49:36.7] JM: A topical question I have for you is, obviously, this stuff around Facebook where they're really getting harangued by Washington around the fact that they allowed Russians to run ad campaign for political stuff that was going on in America. This is being categorized by people in Washington as hacking. It's really more social engineering. It's purchasing ad buys, but there seems to be a lot of confusion around that. We won't really get into the political stuff, but I'm curious, how should companies start to tackle this? Maybe this is a question for an ad operations person, but it does seem like kind of a security problem when you're trying to think about, "Okay. We now need to consider the type of malicious ad buys. That might take place on our system. Do you need to run ad buys through some new kind of set of infrastructure, or how are you thinking about that?

[0:50:44.0] AK: I think it's a really, really tough problem, because for the following reasons, right? You mentioned Facebook, and I don't know like anybody there or I don't have like people that I know work there, but this is my, I think, own opinion about it. Facebook, I think, has millions and millions of advertisers. That alone makes the problem hard to detect.

If I'm a bad guy or I'm someone trying to run — I don't know, a political ad campaign, as you mentioned in that platform, most likely I'm going to slip through the cracks, because I'll would be doing some of the basic, actually, techniques to hide myself, like run — I don't know, like 10,000 campaigns. Each one of them is one dollar. All, basically, the triggers or the knobs Facebook probably have in place will not trigger, because it's too low for an amount. That is not even worth their time to investigate. But guess what? All these like small campaigns are actually one campaign, and you gather all of these dollars and now it's like a hundred thousand dollars campaign. That's one.

The other thing I know about ads and ad review is the problem of ad cloaking. I don't know if people are familiar with this concept here, but what bad people do, or actually I think Facebook or Pinterest also like does this. When an advertiser comes in and they say, "Hey, we want to promote this content." So usually what you do, let's say, for like big advertising campaigns or

campaigns that bring a lot of money, you'll have a team of humans review these ads. So they go basically and go click in that content and see where that competent takes them.

Ad cloaking basically routes you to some content if you are clicking from one IP and another if you're clicking for another. If you're reviewing from Pinterest office or Facebook office, these guys know the CIDR or the IP block of that company. So — I don't know, like a healthy avocado ad will go to the avocado site, but then out in the wild, if a user from Facebook are Pinterest clicks on that, it's an IP block or a CIDR different from the corporate networks of Pinterest and Facebook. It will take you to a weight loss [inaudible 0:53:03.8] or something. So it's a huge problem.

I think people usually try to kind of Ike outsource the ad reviews to different companies, so you change your IP's and you distribute your IPs ahead of these attackers, but these guys also know what are the companies out there that review ads, almost like they knew their CIDR blocks. So it's huge problem. I think — I'm not an expert in the domain, but I know like from my lens, from the security lens, that it's a tough problem and I don't think Facebook intentionally let like this political campaign run. I think it just slipped through their advanced detection, whatever, or systems, because it's a tough problem.

[0:53:45.2] JM: Definitely. Really interesting points. One more question and then, yeah, people start thinking about questions to ask. Amine, so you worked at Apple before Pinterest. I do think of Apple as one of these companies who probably has security down pretty well. They've been at it for a while. How did your experience at Apple differ from Pinterest security-wise?

[0:54:09.6] AK: a huge difference. I think there is a wide spectrum out there in security. So there is the lock everything, block everything, everything is an access request. No one knows what's going on, even like the guy sitting next to you in a different office.

[0:54:31.6] JM: At Apple. Nobody knows what's going on.

[0:54:33.0] AK: There is that spectrum. I didn't say Apple. There is the other end where everything is open, you don't control anything. So that's security. There is the middleware, as I

mentioned, and I think that's the case of Pinterest, where things are locked and others are open and we're monitoring and kind of logging and seeing, making sure everybody's responsible.

Apple tends to be towards like the lock everything. Everything is access based, request based access. Everything is locked. I think requests take like two or three weeks to get honored for you to get access, and so basically they're in the, "Hey, do not enter front," and then once you're in, they don't really do much in terms of monitoring or logging or whatever. They basically give you a hard time to get in. Once they kind of like run all the background checks they want, you're in.

I don't believe in that system and I don't think security should be done that way. I think security should be, I think, a combination of, yes, lock things down, again, like protective crown jewels or things that you cannot afford to lose or things that can be a huge risk to your business. By the same time, have this freedom of moving fast, building things, breaking things, but at the same time have the right monitoring, alerting and auditing in place in case something happens. So that is, I think, the model I believe in. That's what we try to build at Pinterest and that's how Pinterest security program works.

[0:56:08.1] JM: Okay. Questions?

[0:56:10.5] Q: I assume you guys have a service-oriented architecture Pinterest. How do you do interest service authentication to make sure that your services know what each other are and they're talking to the right person?

[0:56:19.0] JM: Question is; how do you do inter-service communication between service-oriented architecture?

[0:56:25.4] AK: It's a great question. So I think we have service authentication and the secure environment. So in other environments we're not there yet because, as you probably know, it's really hard to debug and maintain. But at a high level, what we do is we basically use TLS. We're not reinventing the wheel. We're doing like server-side, client-side authentication. So basically, let's say, you have two services trying to talk to each other. Most likely, each service is provisioned by a separate CA, and then basically they trust each other, basically, because it's

signed by the CA that I trust and you are signed by a CA I trust. Basically, that TLS work its magic. So that's how we do things in our secure environments.

I'm aware of a lot of other efforts out there. I think OPA is one, open policy agent. I don't know if you guys are familiar with that. I think they're looking at the problem kind of similarly, but differently as well, where they have a centralized, I think, service and then agents running in its service and then each agent is kind of like talk to that centralized service. We're going to explore that in 2018. I think it's an interesting idea, but I think, yeah, plain TLS also does the job in my experience.

[0:57:44.8] JM: All right. Amine Kamal [inaudible 0:57:46.5].

[SPONSOR MESSAGE]

[0:58:00.6] JM: If you are building a product for software engineers or you are hiring software engineers, Software Engineering Daily is accepting sponsorships for 2018. Send me an email, jeff@softwareengineeringdaily.com if you're interested.

With 23,000 people listening Monday through Friday and the content being fairly selective for a technical listener, Software Engineering Daily is a great way to reach top engineers. I know that the listeners of Software Engineering Daily are great engineers because I talked to them all the time. I hear from CTOs, CEOs, directors of engineering who listen to the show regularly. I also hear about many newer hungry software engineers who are looking to level up quickly and prove themselves, and to find out more about sponsoring the show, you can send me an email or tell your marketing director to send me an email, jeff@softwareengineering.com.

If you're listening to the show, thank you so much for supporting it through your audienceship. That is quite enough, but if you're interested in taking your support of the show to the next level, then look at sponsoring the show through your company. So send me an email at jeff@softwarengineeringdaily.com.

Thank you.

[INTERVIEW CONTINUED]

[0:59:28.5] JM: The last interview is with Evan Johnson, who is running security at Segment. Evan, welcome to the interview.

[0:59:37.5] EJ: Thank you.

[0:59:38.5] JM: Segment is a product that gathers customer data into a platform for analytics. It's an API. It's also up platform. Give an example for how Segment works so people can get familiar with what we're actually talking about.

[0:59:53.4] EJ: Sure. One big use case for segment is a lot of marketing teams have lots of different end tools that they use where they may use Google Analytics or HIP analytics and lots of different tools. Segment supports over 200 of these end tools that marketing teams might want to use, and instead of sending the data to all of these different places, you can just send it to Segment and Segment will manage sending these to all of the different end tools you want to use.

Another big use case for Segment is data warehousing. So when you send hundreds of thousands of API calls to Segment, we can take them all and massage them and with them in a data warehouse for you to use and your marketing team can go and run SQL queries manually and kind of really dig into your data as deep as you want to go.

[1:00:52.0] JM: When I was preparing for this, I was talking with you over email and kind of asking you what you're working on at Segment. It sounds like a lot of the stuff that you're working on, and this is your word, is boring stuff, and I think what you mean by that is that this company grew like a weed and now you just have sprawling infrastructure everywhere. You were telling me earlier that your usage of the Amazon Container Service, ACS, is going up against rate limits, which is kind of crazy to think about that you're getting rate limited by Amazon for their container service.

So in this crazy high-speed environment with high throughput, you're just trying to basically get things under control and implement some boring security policies. So with that in mind, tell me

something that would be easy to implement, would be kind of boring if not for the scale of segment.

[1:01:46.6] EJ: Sure. One of the — I think while I've been at Segment, I've been at Segment for about eight or nine months kind of as a cloud security lead, and a lot of the things that I've done are kind of all over the spectrum of security best practices. So kind of one of the first things I did was helping get a good story around secrets management, and from there I moved on to kind of a lot host-based hardening where now we are running go-audit on all of our production clusters and machines where any command that an engineer runs in production gets siphoned off and stored into S3 for us like analyze and keep forever or for some retention period. From there, I moved on to now just kind of AWS least privilege access for employees, and Segment kind of was just about to hit this hyper hiring phase when I joined. We're just about to raise a series C and kind of now, like you said, we're up against all these rate limits with ECS S3. There is another —

[1:02:58.9] JM: Rate limited by S3?

[1:03:00.5] EJ: Yeah, it's pretty bad. We're battling rate limits on almost every — Oh, parameter store, SSM. We're battling rate limits, basically, everywhere. U.S. West two is very unhappy with us. Yeah, a lot of the boring security stuff —

[1:03:18.5] JM: Okay. When you get rate limited by S3 —S3 is like your canonical data store for stuff. So when you say — Like when customer makes requests to store their data on Segment and Segment makes a request to store the date on S3 and S3 says, "Sorry. Not right now." What do you do?

[1:03:41.9] EJ: Yeah. So retries are bit, but the — Luckily we haven't run in this issue with like production customer data coming in. It's been — We actually store that in a couple different places, and so we haven't run into the problem where it's a customer data. S3, we hit rate limits very briefly on. It hasn't been our biggest problem at all. It's been mostly ECS. Yeah.

[1:04:07.6] JM: Okay, bad question. But you've hinted at a couple of things that Amine was actually talking about with Pinterest, and the first one, I guess, kind of service sprawl and like

how do you manage authentication between different services. The one was, essentially, the key management. How do you manage the — Sorry, the secret management, more broadly. Why don't you explain what secret management is, this whole discipline of secret management and then explain specifically why that has been challenging in Segment?

[1:04:41.9] EJ: Sure. So one of the major — Yeah, so secrets management is you don't want to have your secrets baked into your source code. Obviously, this is really bad for a lot of reasons, like Amine pointed out. they'll end up on GitHub. GitHub repository ends up being made public, and people can — Like attackers who can go through Git history and find secrets in these GitHub repos, and you really need a central place to manage your secrets, because in addition to it being bad and to find secrets in source code, it's also good to have it in a central place, because you can roll them and manage them and manage access control to them.

One example, if you have a Stripe API key baked into source code in 50 different locations, like how do you even roll that? It's just not really even possible. It's like, "Okay. Everybody type Git commit at the same time, and here's the new key and this is going to work, but we just all have to press enter at the same time." It's just not really feasible unless you have it in a central location where you can roll it and manage it. Some of the properties of a good secret store, you want encryption to store your secrets at rest. You want strong authentication and authorization to the secret store. So you don't want all the secrets, all the services to share the same secrets, and to do this, you need like a strong identity-base.

In AWS, the biggest secrets are — Well, the identity base is all IAM, so that's like the — In AWS, you'll go crazy trying to do anything, but IAM for your identity base. Amine kind of talked about this with AWS keys. Keys are like the crown jewel of hacking any AWS based company, because they are just impossible to manage, really, as your company scales from like 40 engineers to 80 engineers, to 160 engineers, and then 320 engineers. That's just so many keys. Really, just as few keys as possible. There're only a few API calls you really need to have a key for. The rest you can use task roles or roles, whatever it's running as.

Employees even were moving towards a world where employees don't even have AWS accounts. It's all through Okta and single sign-on SAMLs. Those employees don't even get keys to have their access to our AWS environment.

[1:07:29.4] JM: I think another thing that you'd hinted at there is — One thing you want out of your secret management system is dynamism. You want to be able to roll your keys. You want to be able to change your keys. The problem that you mentioned with, if you have a Stripe key hardcoded in 50 different places and then you go in your Stripe dashboard and say, "I want change my Stripe key." That's going to invalidate it for all 50 people who are using Stripe for different things across your company. So what you would need to do is go to all those 50 people and say, "Okay. Hey, all ones, we're going to need to change to using this new key," but if instead you just have all of those 50 people pointing to a dynamic variable that is hosted on AWS somewhere, and then you can change that dynamic variable, that can make things a lot a lot safer.

[1:08:27.5] EJ: Yup. This is also — Like rotating keys is also a problem that basically every software as a service company wants to solve for their customers. How it works on Stripe specifically is I believe they have a four hour window where you have to keys for four hours where you rotate the old key an startup services with the new key, and both keys work for that four hour window and then they revoke the old one at the end of it.

[1:08:53.2] JM: I guess a different topic, Segment is a pretty flexible platform, do you ever have issues with customers who abuse the product? Do something that wasn't intended?

[1:09:06.3] EJ: I don't know how to answer that. That's pretty abstract.

[1:09:09.5] JM: No. Well, things that are maybe illegal or they like sign up for a bunch of accounts in order to use — You have a premium platform, right?

[1:09:19.9] EJ: I believe you have to pay. I think maybe we have a free tier for up to 1,000 monthly tracked users, but we tend to push people towards paying very quickly, and I think even for free, we ask you for a credit card. Actually, I think this is one of the big things that helps kind of prevent fraud and abuse. It's like we want high quality customers who will be able to pay us.

[1:09:48.3] JM: So you just do not — You don't have an abuse problem.

[1:09:50.9] EJ: It has not become a problem. If there is one, I don't really [inaudible 1:09:56.6].

[1:09:57.5] JM: Okay. Cool. With Nick earlier, we talked about, basically, how a webpage gets secured, the basic ideas of HTTPS through a certificate authority. How does it differ when we're were talking about an API? Segment is essentially an API product. How does securing an API and expressing to the customer that the API is secure? How does that differ from securing a webpage where you see that beautiful green lock?

[1:10:31.8] EJ: Yeah. One of the things about api.segment.com where we ingest messages from our customers, API is like super simple. It's maybe a couple hundred lines of code. It's an ingest pipeline and it receives a message and it puts it in a queue and then those messages get acted on later. So it supports one supports one HTTP method, like if you send a post, we handle it. If you don't send the post, then we would just drop the message.

I think actually this property that is important is being opinionated. Like we only support HTTPS. We only support one HTTP method post. We only have a few API endpoints and being really opinionated and driving your customers to the few things that they should be doing, I think, goes a long way.

[1:11:28.4] JM: How do you audit the segment infrastructure for security holes?

[1:11:35.2] EJ: I've been looking at it very holistically where I look at all of our clusters where I look at all of our API key usage, and I've been looking really top-down and everything we have, but if you mean auditing, like monitoring and —

[1:11:53.4] JM: I just mean like diagnoses. I don't really know what that word means. I'm just using it, and I've heard it used elsewhere. Well, Amine also was talking about, basically, this idea of incident response as being closely tied to your monitoring and your logging infrastructure. I guess I'm wondering how, from your point of view, monitoring and logging fit into security. So you hear about this dev sec ops term more recently, which I think is kind of the idea that in order to do security effectively, you kind of have to have this new role of security to have a person that really understands infrastructure and monitoring and logging and that new HIP term observability, where observability is how do I understand how the different pieces of

infrastructure in my company are even being used. I've got all these services. They're communicating with each other in different ways, but can I actually draw a map of how those things interact with each other? That's kind of the dev sec ops thing. I think it's like, "Okay. We need to do this, because the only way to actually understand the security of our infrastructure is to understand like observably how the infrastructure interacts on a regular basis." How does observability fit into your work?

[1:13:19.8] EJ: That's tough. You're saying stuff about dev sec ops, and I was getting excited about some of the stuff I've been working on there, but I'm not really sure how to —

[1:13:28.0] JM: Well, then tell me what dev sec ops means to you, because maybe I might just be totally overloading that term.

[1:13:33.6] EJ: No. words are hard. Observability is — I don't really — I don't know. Dev sec ops to me and security to me in the cloud, before — Segment is the first company I've worked that that's all cloud-based and, really, the first company I've worked at that has anything in the cloud. So it has been amazing to me to be able to write a lambda function or like mini-lambda functions that inspect many different parts of our infrastructure or what is happening in our AWS environments, and I can get signal back about positive and negative things that are happening in them.

I was talking about AWS keys. Every four hours I download every credential report, AWS issues, meaning every segment account, AWS account. Every hour I download every version, every ECS task that is running in our AWS environment so I can notice anomalies really easily, so we know when new keys are issued, so we know how many keys are in existence in every single account.

I think it's really — In the cloud, it's really about finding your signals and driving the numbers and the data points towards the direction you want them to go in. Like, obviously, having 200 AWS keys is really bad, and so I have this continuous monitoring of all of our keys with lambda functions, and I can just drive this signal in the direction I wanted, which is down. And so I say, "Why do we have keys.? Let's go get them."

[1:15:13.8] JM: That's such a cool application of AWS lambda. I hadn't really heard of that before. So I'm just imagining, you're writing the scripts and you're deploying them to AWS lambda, which is a function as a service platform where you can basically deploy these ad hoc, totally stateless scripts and just have them run and have them do something and then spin down when they're done. It sounds great for kind of orchestrating these one-off jobs that you want to run on a regular basis to inspect the health of various aspects of your infrastructure.

How many of those do you have going? Because I'm just kind of like imagining you sitting at your computer just like writing these random one-off scripts in a very disorganized fashion, but are you starting to think about like turning this into a more process-based? I don't know. Something you're going to put in the wiki where here's — If you want to write a security script, here's the way you do it and deploy it and send email updates to everybody. Are you going to standardize that?

[1:16:15.3] EJ: I have thought about open sourcing it, but I really want to rewrite them in Go in January when you can write lambda Go functions, which I think January is the release date for lambda Go, and so I'm really excited about that.

Yeah, we have a repository internally of all of our security lambdas. There's about seven of them, and I just write them quickly as they need or I notice, "Hey, this is good signal. We can gather from a lambda function," and I've just been whipping them up. AWS has actually been introducing a lot of new products that really help — So I've heard people at AWS say they don't want security to be expensive, and that is, I think, pretty true, like they have these products trusted advisor, which you can pull down reports about things that they notice, like public S3 buckets and public — And like ways you can save money. There's like a lot of things in a trusted advisor report. They've got credential reports, and then they've got all these — Like I said, this is my first time working in the cloud and it's amazing that there's an API already that tells me every instance running in our environment, every key, every task, every service when the service was created. It's pretty incredible. Then IAM is free, and it's pretty cool.

[1:17:47.2] JM: And yet we were talking before this interview started, you said you're thinking about going multi-cloud. So I'm assuming with multi-cloud, you're going to sacrifice some of the kind of beautiful unification of being entirely on AWS, because you have — IAM is identity and

access management. This is a thing that is an Amazon cloud notion. If you want to go and say, “I want to also do things on Google cloud.” You're going outside of your walled AWS garden and you have to start to think about, “Okay. So if I'm going to orchestrate some kind of workflow between AWS services and Google cloud services, I need to figure out how am I doing that translation of identity between those two platforms, for example.” So maybe you could explain why you're going multi-cloud and are you adopting any kind of best security practices around that?

[1:18:41.8] EJ: Yeah. So we've been using GCP for some of our newer products that we’re launching, and specifically some of the offerings GCP has is really compelling, like BigQuery and Pub/Sub. BigQuery is just amazing, and then Pub/Sub is super cheap and really solid.

I think for the multi-cloud strategy, Amazon is releasing Kubernetes support, and Google has been working towards Kubernetes support and they’re a huge sponsor of the cloud native foundation and they've been investing super heavily in Kubernetes.

What I think is what we will probably do, this is just me guessing, is there's going to be some time where we wait for the dust to settle and we can kind of like, in the words of one of my coworkers, we can sit on our hands for a while and really think about what we are going to do and the best strategy for tackling this problem. Until then, when we can figure out how Kubernetes would work on multi-cloud and how identity will working all of these problems that we’re going to have to tackle, I think once it's clear how are going to do it, then we’re going to go all in, is my impression.

[1:20:02.8] JM: So you can use EKS, the Amazon Kubernetes Service. You can use that to go home multi-cloud. What do you mean by that?

[1:20:11.6] EJ: Oh, no. I'm not sure if we’ll use EKS. I’m not sure if — I do know how it all — How the pieces will fit together.

[1:20:21.5] JM: Okay. Right, this is the dust settling.

[1:20:22.3] EJ: Yeah. I think we're going to wait and see, because there's some other compelling Amazon services that are coming out the far gate or just containers, which people —

[1:20:32.9] JM: Container without the orchestrator.

[1:20:34.9] EJ: Yeah, so we don't have to run EC2 instances. We can just have containers. There's a lot of compelling stuff, but I think for now it's just like wait and see until we have a really great plan of attack, and then I think that is what will end up happening, but I have met with people at Google and we've talked about secrets management and stuff in Google cloud, and this is something that's like hot on what they're thinking about as well.

[1:21:06.1] JM: The multi-cloud stuff?

[1:21:07.9] EJ: The multi-cloud stuff and then the security of how will all of these fit together of, like, will there be an identity exchange, where I can trade an IAM role for a GCP project?

[1:21:21.0] JM: I'm sure Google would love that. I don't think Amazon would be so compliant.

[1:21:24.4] EJ: I don't know. I have no idea how any of that will work or what will happen, but it should be pretty interesting. I'm pretty excited for multi-cloud security and how all these stuff is going to work.

[1:21:36.6] JM: Okay. I'll ask you one more question, then we'll go to audience questions. So you used to work at Cloudflare and now you're at Segment. How does the security at Cloudflare differ from that of Segment?

[1:21:50.9] EJ: I just looked over at Nick who's at Cloudflare. This is actually a pretty interesting question. So I think Cloudflare, every single like company meeting or company meeting, all hands meeting, they really do come up and say, "This is a security company, and it's important," and this resonates with every engineer. You can tell every engineer you talk to, if you try to do something crazy, anybody will slap your hand. At Segment — Segment, I joined, they're a lot smaller than Cloudflare when I joined, and it's also a completely — It's not a security product,

and so a lot of people are coming to this — Coming to reckoning that if segment is going to be a big enterprise, security is super important.

Now, the difference I think is now I'm at a point where I'm actually instilling those security values into the entire company and I'm evangelizing it with all the engineers. So security-wise, it's like things come a lot easier at segment, because we're all cloud. We can just like, "Okay. But it." We have security groups. We have IAM. We have all these policies that we can set in Amazon and it's just an API call away. At Cloudflare, it was a lot more work, like a lot more work. To even get much smaller projects through the door for security and — But I think Amine mentioned this, the different models of security, where they'll slap your hand at Apple. Yeah, at Apple you have to wait a couple of weeks and then we finally get an access request, and that's how it felt at Cloudflare sometimes for a lot of people when they need access to something, and you really have to fight through these policies and whatnot.

At Segment, since one thing I've been working on is being able to control the access of all employees through Okta. They have this really slick management, so management layer through Terraform and you can control what roles every employee has and exactly what policies and permissions every single employee has. I mean, that kind of thing would take years to even get planned at a company like Cloudflare.

I think if I had to sum it up, it's just is not comparable, because one is cloud. You can buy it. There's so much stuff already there, and then one is all running their own infrastructure. There're 115 data centers or however many number there is now, and so it's really just totally different challenges and it's — I don't know. They're both really cool environments.

[1:24:53.1] JM: Make sense. Okay, audience questions. Anything you're concerned about coming up in 2018 security-wise?

[1:25:00.0] EJ: I don't know. Security-wise, it's like look at the price of Bitcoin. That's like kind of crazy. People are just going to be stealing these things still. I think crypto currency highs will accelerates. But like security-wise, I think it'll be the usual where — Yeah, two to three — I think there'll be, just like every other year, there'll be like two or three major bugs that come out and it

will be a big news for a week and people patch and life go on. But, yeah, I don't really see anything major, major, major.

[1:25:37.6] JM: What kind of security gaps are there in multi-cloud today?

[1:25:43.5] EJ: I really don't know. I think nobody's really doing it well yet. Maybe the biggest of enterprises really understand this, where they have like tons of data centers where they have like their own infrastructure. They've got stuff on AWS. They've got stuff on GCP. I really don't know what the gaps are, but we'll know when we get there.

[1:26:07.2] JM: What is a lambda and what is the advantage of building some security infrastructure on AWS lambda?

[1:26:13.2] EJ: Sure. I think a lambda is a function that you write that Amazon will run, and I think they support four or three languages; Python, Java, .NET and JavaScript, and coming out soon is Go Lang, which I am super stoked about, and then there's a few, maybe they announced another language. I'm not really sure. I really care Go Lang. But you just write this function and Amazon will run it and you can set alerts. So you can set a function, or you can set triggers to run this function, not alerts.

One common one that I generally do is once hourly or once every four hours or so, and for running something once hourly, it is incredibly inexpensive. I think it's like per million invocations of a lambda. It's may be a dollar or something. So like there's no additional cost to run this — Like comparing a lambda versus an EC2 instance or a Docker container, if all you need is code and no infrastructure or anything, it's really just super convenient to write this function, store the result in S3 or in whatever relational database, wherever you want to put it and not have to think about where is this thing going to run. I've got of set the cron to trigger this thing every so often. It's just super convenient.

[1:27:50.2] JM: Explain the catch. What's the down of using serverless?

[1:27:54.7] EJ: What is the downside of using serverless?

[1:28:00.2] JM: There's the cold start problem, because these functions as a service are running on some random piece of infrastructure on AWS and you're just scheduling whatever function that you write against AWS's infrastructure and they're like, "Okay. When we have some random infrastructure available, we'll schedule it." At least that's kind of how I understand it. They probably have some containers, but that's the one thing, is the cold start problem. But then there's also — Yeah, just kind of — You don't know how long it's going to take until your function is going to run and then, also, you can't really use these for — Most people don't build kind of apps where you have state managed in the container itself, but with function as a service, you really don't want to do that, because these are very nondurable. You just expect that these functions can fail at any time and maybe you want to retry them or anything, and we do kind of make those assumptions about our containers or our VM's or whatever, but you really have to make that assumption with the function as a service.

[1:29:02.8] EJ: Yeah. That is a downside. Cold start, I think, most of the stuff is stuff that it can run an hour and a minute or an hour and five minutes and I won't really even know the difference for all of the security monitoring and just getting great baseline security metrics that I'm looking for. That isn't something that I really care about, and I think the statelessness of lambdas, you can always use one of their RDS or something from — I've never done this, where I needed a stateful lambda since I'm mostly just writing reports into a place where I can continuously monitor the latest one. I've really had a great, great success with the lambda functions that we wrote, but I think Amine had something to say.

[1:29:55.9] JM: Good luck debugging them. Yes, that's another downside. On the AWS website, [inaudible 1:30:02.9] you can just export everything to cloud watch logs? No. Okay.

[1:30:08.3] EJ: Just write perfect code. Come on.

[1:30:10.9] JM: All right. I think that's everything. So we'll wrap it up there. Thank you, Evan.

[1:30:15.3] EJ: Thanks for having me.

[END OF INTERVIEW]

[1:30:24.6] JM: GoCD is an open source continuous delivery server built by ThoughtWorks. GoCD provides continuous delivery out of the box with its built-in pipelines, advanced traceability and value stream visualization. With GoCD you can easily model, orchestrate and visualize complex workflows from end-to-end. GoCD supports modern infrastructure with elastic, on-demand agents and cloud deployments. The plugin ecosystem ensures that GoCD will work well within your own unique environment.

To learn more about GoCD, visit gocd.org/sedaily. That's gocd.org/sedaily. It's free to use and there's professional support and enterprise add-ons that are available from ThoughtWorks. You can find it at gocd.org/sedaily.

If you want to hear more about GoCD and the other projects that ThoughtWorks is working on, listen back to our old episodes with the ThoughtWorks team who have built the product. You can search for ThoughtWorks on Software Engineering Daily.

Thanks to ThoughtWorks for continuing to sponsor Software Engineering Daily and for building GoCD.

[END]