EPISODE 993

[INTRODUCTION]

[00:00:00] JM: Large software companies have lots of users and the activity from those users results in high-volumes of traffic. These companies also have a large surface area across the enterprise. There are hundreds of services and databases that are fulfilling user requests across a large enterprise. Take a company like Airbnb. As the requests enter the infrastructure of a company like Airbnb, the requests will travel through the different services and results in database queries, payments and other transactions and will finally result in somebody booking a place to stay.

These transactions also result in the generation of log messages. The log messages tell the story of what is happening across the entire infrastructure at the company. Log messages can provide valuable data for security and site reliability engineering, but analyzing a high-volume of log data requires a scalable system that can account for that high-volume.

Jack Naglieri is the CEO of Panther Security. He previously worked at Airbnb where he helped develop a system called Stream Alert. At Airbnb, log messages are buffered into distributed queuing systems like Kafka or Kenesis, and then they're written to bucket storage systems, like S3. These logs are processed by AWS Lambda functions that test the log messages for rules defined by a system operator.

Jack left Airbnb and started Panther Security to generalize the tools that he built within Airbnb and built a company around the same ideas. Jack joins today's show to discuss modern logging infrastructure, his work at Airbnb, and his experience building Panther.

[SPONSOR MESSAGE]

[00:01:56] JM: The start of the New York is a great time to evaluate your career. It's a time to consider your salary. It's a time to consider your job or maybe consider changing your entire career path. Seen by Indeed provides a path to new career opportunities while reducing the pain of the traditional job search process. Seen puts tech candidates in front of thousands of

companies like Grubhub, Capital One, and PayPal across more than 90 cities. Just create your profile from your resume and they'll match you to the right roles based on your needs.

Every Seen candidate also gets free access to technical career coaching, resume reviews, mock interviews and even salary negotiation tips to seal the deal. Join today and get a free resume review when you go to beseen.com/daily. That's B-E-S-E-E-N.com/daily, D-A-I-L-Y.

Seen by Indeed is a tech-focused matching platform. If you're ready for a new job, you're ready for Seen by Indeed. Join today and you get a free resume review when you go to beseen.com/daily. That's B-E-S-E-E-N.com/D-A-I-L-Y.

Thanks to Seen by Indeed for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:03:26] JM: Jack Naglieri, welcome to Software Engineering Daily.

[00:03:28] JN: Thanks for having me.

[00:03:30] JM: You are at Airbnb for 2-1/2 years. What were the canonical security problems at Airbnb? What were the problems that you saw on a recurrent basis?

[00:03:39] JN: Yeah. I think it's the problems that every company faces that's going through hypergrowth and it's starting from zero. I mean, you'll see a lot of really sophisticated companies that don't really have really any security program, but they've made a lot of revenue and they've grown significantly and now it's time to get serious about security and like start building this program out.

The canonical problem is always just like how do we make sure that we're covered in all these critical areas and we are gathering the right data and we're really understanding everything that's happening in the business and in our technology infrastructure.

Believe it or not, probably you're really shocked if you knew like the state of a lot of security

programs in the industry for certain apps that we would use every day, for example. They're still very immature in their security programs.

[00:04:32] JM: In what way? Say more about that. In what ways are they immature?

[00:04:35] JN: It's such an understaffed field. A lot of people just fall behind naturally and the business isn't prioritized until they really need to. Maybe they're getting to go public, and that's generally the biggest catalyst that I've seen, or the company operates in like a highly regulated industry, so they need to care about it early on.

But generally, it happens in the later stage. You'll see a security team start to come online and they're like, "Okay, what's our engineering org doing? How is our cloud environment set up?" I'm speaking more for like modernized, like cloud-age companies. More monolithic larger companies that really have like super established security teams that are probably like in the hundreds, like Facebook, Google. When I worked at Yahoo, it's like that as well. We had like close to a hundred people. But for these more modernized hypergrowth unicorns, it takes them generally a little bit longer because it's not something they prioritize.

Actually, when I joined Airbnb, it was kind of like that, right? I joined in 2016 and Airbnb was just like starting to become more mainstream and was really picking up, and that's when they have like the $25 billion evaluation and they were kind of like speeding into the mainstream. I guess as the business, they hired the director of security who started hiring all of these other managers to say like, "Okay. We need to focus on this and this, and we need to start doing a gateway for our compliance." It's just like the normal things a company does to prepare for an IPO.

[00:05:56] JM: At that point, their software was, I imagine, a highly scaled Rails app that was held together with duct tape and chicken wire and they're just trying to figure out simultaneously how to keep the lights on and how to add some better security practices.

[00:06:14] JN: Yeah. I mean, the company had a lot of established engineering practices and procedures and they had built all these great internal tools and it was a really impressive team, to be honest. Every hypergrowth team, as you say, they kind of make what they have worked for

a really sustained amount of time and then they kind of go back and fix it. Twitter did that as well, right? Twitter rewrote their whole platform from a Rails app into like Java microservices. I'm sure Airbnb is probably doing the same thing now. It's kind of like that with security too. You do all these things and then you kind of go back, clean it up and then you prevent anything bad from happening in the future.

[00:06:51] JM: What were the different areas of security that that security executive hired into? What were the security teams that he established when he was forming those security teams?

[00:07:02] JN: It's the same teams that you would see in any organization. You'd see people focused on application security. I feel like with Airbnb, that happened early on. There were actually people who were doing application security, from an earlier state of the engineering organization's life. Then as time went on, they actually split them out into like a proper security org. People always cared about AppSec early, because it's something that is really tangible and like engineers can see without the presence of a security team like, "Okay, we need to care about validating input into a SQL database," and things like the obvious things, right?

Then as the org scales, it's more like, "Okay, let's think about dedicating people towards compliance and cloud security and making sure that we're upholding like CIS and if we have standard sort of PCI that we have to uphold, making sure that we focus there." Then moving more into incident response, which is the field that I specialized in when I was a practitioner. It was like, "Are we collecting the right data? Do we understand what's happening in the systems?" and getting an understanding of what normal is. You just start to focus on all these different areas in security too, like product security. People focus on the application itself.

Then you have like incident response and then you have compliance. I think those are probably the main ones and there's like a bunch of other areas you can focus on. It really depends on like the nature of the company and the product and what your assets are, like data securities and other really popular one too.

[00:08:31] JM: Tell me about the incident response side of things. How does an incident get discovered? What are the data streams that contribute to discovering that incident and how does an alert get created? How does it get propagated through the organization?

[00:08:47] JN: Yeah. It's a good question. There's a lot of different ways. I mean, the basis of good detection and response is really having a rich set of data that will essentially let you know what's happening in all these different important areas and you use that data to create rules to say, "Okay, we expect from our logins," for example, "we expect that only engineers should be able to log-in to our systems. We expect that certain files should exist in certain places."

You can kind of like define and patterns what normal is. Then when anything outside of that happens, you get some ping to your team. Whether it's like a Slack, or a PagerDuty, or whatever. You flag to your incident responders, like, "Hey, something suspicious is happening." You can do rule-based detection. You can do machine learning, AI. You can do more like cutting edge stuff, which in my opinion is probably not the greatest place to start just because of the false positive rate, but just kind of like a personal opinion.

Then we have indicator of compromise, like you can buy IOCs from certain vendors, like Secureworks. There's open source IOC's you can use. An indicators of compromise is basically if you see the presence of an empty MD5 hash, which is like a file hash on one of your laptops or one of your servers and you might have this type of malware which would lead to this type of compromise. You can do pattern matching like that, which also can be highly false positive prone. It's essentially the combination of all these techniques, is what goes into like a well-rounded detection program.

You'd look at indicators. You use may be some AI and ML. You do some deterministic-based rules. A lot of deterministic-based rules I think are based off of like the company's specific business logic, because everyone infra is different and everyone's going to have like slightly different limitations of how their engineering team works and what normal is for them. I think it's important to like build that into whatever tool that you're going to use for detection.

Historically, the way this is done is – I mean, the most popular way right now in my opinion is people just using likes Splunk or Elasticsearch and kind of just feeding it into there and then building on top of it. More historically, it was within a SIEM, which is a security information and event management. You would feed all of these log data from like your laptops, your servers,

your networks. Any of these helpful contexts that could be used for security detection, you'd feed it into this box or this appliance and then it would tell you what it thought was bad.

But the issue with that, and I think you pretty rarely see that in Silicon Valley companies, like in higher tech companies. The issue is that they have trouble scaling. They don't provide enough flexibility in how you want those detections to work. They're highly opinionated. I think as an engineer here in the valley, people want to be empowered to create these detections themselves.

Actually, what led to like us building StreamAlert at Airbnb, like that experience really shaped how we wanted to think about it going forward. When we joined Airbnb, I was brought in as an engineer, but my manager at the time, he had also come from a larger tech company. He was from Facebook. I was from Yahoo. Then someone else was from like Dropbox, and we had all these experience with like with scale and knowing what works and what didn't and we knew we wanted to do something fresh and kind of take all these learnings that we had from like the old world and kind of like bringing it into something fresh and new.

[00:12:17] JM: Let's talk a little bit more about the tooling that you build at Airbnb. Aribnb has a high-volume of traffic that's coming through it and you built a tool at Airbnb called StreamAlert to deal with the classification analysis and alerting based off of that high-volume of data. Explain what StreamAlert does.

[00:12:44] JN: Yeah. StreamAlert, the way I always described it is a serverless real-time data analysis framework, and we released it at Enigma Security Conference in 2017. It's essentially a way for a small team to do security monitoring in a cloud-based environment at a really high-scale. The challenge that we always had before was like if you want to consume like 10 terabytes of log data a day into Splunk, you're going to go through a lot of pain of just even operationalizing that.

My manager at the time, his name is Jordan McReynolds, he had this idea of utilizing Lambda. He was like, "We can build a pipeline on top of AWS Lambda, which is a serverless service from Amazon," and the way that that works is that you write the application code. You upload it to Lambda, and then Lambda runs it for you.

Serverless has obviously taken off in the industry and it's extremely popular now. But at the time we were like, "Yeah, we can really use this for a security application and we can accomplish the monitoring problem while using Python instead of a DSL," like the Splunk DSL or Apache Lucene, which is what is powered by – Or Elasticsearch uses for its search language. We wanted to really provide flexibility and scale into our monitoring stack, and that's really what the basis of that project was.

[00:14:11] JM: What purpose did Lambda serve?

[00:14:12] JN: Lambda hosted the application itself. The way that StreamAlert works is that it's a collection of Lambda functions that serve a certain purpose. We have a Lambda function that would do like classification of data to say, "Okay, this is an OS query logger. This is a cloud trail log from AWS." Then there'd be another Lambda function that maybe sends the alert out to your Slack, or your PagerDuty, and then there's another Lambda function that helps clean the data up and things like that.

It's essentially a way to create the different microservices that collectively are an application. You can pass data through all the different Lambdas and you can mix and match them in all these different ways. You can connect them with queues. I mean, it allows you to build like a more complex architecture with little overhead, and that's the way they benefited.

[00:14:59] JM: Help me understand this. Let's say a user is buying a reservation on Airbnb and when they buy that reservation, the request is going to propagate through a bunch of backend services. Those backend services are all going to create log messages. Those log messages presumably are going to make their way through StreamAlert somehow?

[00:15:23] JN: I can't really explain in that level of depth how it works because of operational security.

[00:15:29] JM: Oh, sure. Okay.

[00:15:29] JN: I mean, the way I'll speak more generally about it is there's a lot of different data sources that we collected internally, and it really depends on what your goals with monitoring, right? It's like do you want to detect application-based attacks? Because if so, then yes, you would want to take an approach like how you just described. But there is a lot of other rich data sources that have data generated from that same event.

For example, if someone just generating traffic to the platform or as a result of that the application is doing something, that generates a lot of data as well. Maybe to take it a step further, you would say based off of a request for a reservation or for a booking, that generates an actual database call. We can monitor the database calls too. We can say, "The database call came from this application and it ran this query, and we know that that's normal and that's expected." But if for some reason a database query was issued from some random place, that's the type of activity we want to monitor.

I think the application level of context are probably more specific only for application security engineers, because for me as an incident response engineer, I don't really have the full context of like how those requests should be handled and what the responses should be, because I don't have the – I'm not an application developer. I don't have that context. But I can understand at a lower level that database queries should be coming from a certain place. Those are the types of things that we would look at as an incident responder.

[00:17:00] JM: Let's say I want to instrument my database so that all of the queries into that database, the log messages that are generated from those database access requests are going to be put through StreamAlert. What do I need to do to configure that to make it a reality?

[00:17:22] JN: I mean, at the end of the day, and Panther is the same way as this. But you just need to get it into the AWS ecosystem somehow. If it's a native AWS database, like you're in luck, because there's a lot of native tooling that we can just use already to feed it into like either S3 or into Kenesis or something else that could get picked up by our Lambda functions. That's another huge benefit of developing in Lambda is that you're already plugged in to the rest of the AWS ecosystem.

But that's also kind of the downside too. It's like if you wanted to take a database that's on-prem or in a different cloud, you'd have to feed it into AWS in some way. There're a ton of tools that help you do this. You could use Logstash of Fluentd or some type of message bus, and that's fairly trivial to set up. But the whole premise is just like get it into the ecosystem and then we can use Lambda functions to consume it.

[00:18:12] JM: Got it. The idea is log messages generated from some point in your infrastructure and you get the logs shuttled to some place where they're aggregated and organized. Then you can run Lambda functions over them. The Lambda functions are executing. In the case of StreamAlert or in Panther, the company that you've built, executing Lambdas which are, as I understand, Python scripts that are analyzing the traffic for rules that you've created.

[00:18:43] JN: Yeah. In StreamAlert, everything is written in Python. In Panther, everything is written in Go, except for like one Lambda function. But all of the others are primarily in Go lang. When we started the company in 2018, that was one of the first things that I wanted us to focus on. We're doing stream processing, and Go is just a better choice for that in opinion just because it performs better and it's a compiled language and it's exactly typed. There's a ton of safety benefits as well.

But we made the choice to do StreamAlert in Python, I mean, for one, that was really one of the only options at the time. Go lang actually wasn't even supported in Lambda and none of us were Java developers, which was the alternative. It was either like Python, Java or Node. Python seemed like the obvious choice. Also, the industry itself, the security industry is like somewhat fluent in Python. That was actually the reason why we chose that for the rules language. The actual Python rules that you would write as an analyst, it's written as a Python function and you say like, "Given some input, let's do some logic. If we return true, that means we should fire an alert."

Let's say we have a set of office networks and we're monitoring the SSH traffic. So someone remotely connecting to a box, to a server, or something, and we can say, "All of our engineers are in this office network. We should only ever expect the log-ins to come from there." You can basically put in the list like, "These are the office networks. Did someone log-in from outside of

this list? If so, generate an alert." That type of logic is helpful and people understand that, engineers understand that.

The one downside is that you do have some teams that aren't writing any code at all. That can be a little bit challenging. But generally, there're enough examples, and Python is the easy enough language to where you can kind of hack through that and you get the benefit of being able to use a scalable platform.

[00:20:44] JM: Right. This is one thing I find interesting about it, is you're building a security tool that – There's a fair amount of custom configuration that the user is going to have to do to create rules for their specific application what kinds of rules are you scanning your log messages for, because you're maybe trying to scan for log-ins from outside of a certain geo or you're trying to scan your log messages for PDF files because you want to know if PDFs are being open across the organization, because those things result in malware.

The idea is that these rules are going to be written by a technical analyst in many cases. Maybe somebody who is somewhere in between the technical level of a software engineer and like an operations person. Can you tell me more, who is that person? What is that role, this security analyst role who would be writing rules like this?

[00:21:55] JN: Yeah. It's primarily people who focus on detection. Generally the way that this would work is they would just dig through log data and try to pick out patterns or try to identify log files themselves that are "suspicious", and a lot of it is like – I mean, that's generally called like threat hunting. There're analysts who come through data that way. You can use techniques that I was mentioning before where you have lists of indicators and you're continually comparing that against logs.

Then if you get hits there, then you can do more custom-based triage to say, "Okay. We got a hit on an IP address. Some network traffic occurred to this malicious IP address in this random country. Let's go back and look through the logs to see what was happening in the system at that time." That's a common thought process an analyst would have.

The whole goal with StreamAlert and then subsequently with Panther was to automate that process to say, “Okay. We can build in indicators into the detection platform. We can build all this intuition that an analyst would have and patterns it with what I want I look for. We can just predefine it into a Python rule and then we’ll have a system continually scanning logs and they come in in real-time. That’s really like the type of person. It’s just a normal sort of detection analyst, like someone who’s familiar with doing breach investigation and maybe forensics to an extent. But most like just doing threat hunting and incident response.

[SPONSOR MESSAGE]

[00:23:40] JM: Being on-call is hard, but having the right tools for the job can make it easier. When you wake up in the middle of the night to troubleshoot the database, you should be able to have the database monitoring information right in front of you. When you're out to dinner and your phone buzzes because your entire application is down, you should be able to easily find out who pushed code most recently so that you can contact them and find out how to troubleshoot the issue.

VictorOps is a collaborative incident response tool. VictorOps brings your monitoring data and your collaboration tools into one place so that you can fix issues more quickly and reduce the pain of on-call. Go to victorops.com/sedaily and get a free t-shirt when you try out VictorOps. It's not just any t-shirt. It's an on-call shirt. When you're on-call, your tool should make the experience as good as possible, and these tools include a comfortable t-shirt. If you visit victorops.com/sedaily and try out VictorOps, you can get that comfortable t-shirt.

VictorOps integrates with all of your services; Slack, Splunk, CloudWatch, DataDog, New Relic, and overtime, VictorOps improves and delivers more value to you through machine learning. If you want to hear about VictorOps works, you can listen to our episode with Chris Riley. VictorOps is a collaborative incident response tool, and you could learn more about it as well as get a free t-shirt when you check it out at victorops.com/sedaily.

Thanks for listening and thanks to VictorOps for being a sponsor.

[INTERVIEW CONTINUED]

[00:25:30] JM: This kind of security tool where you're giving the user the freedom to engineer their own rules that are going to be evaluating log messages, or just evaluating data across a platform if we're to generalize it. Can you help me understand, how does this fit into the mindset of a security buyer? Do security buyers expect a product that requires them to do some programmatic work, some configuration work and so on, or do they want to just buy something that's like off-the-shelf. They take it off-the-shelf, they install it and it magically solves their security problem? I haven't done a ton of coverage on security products. So I'm just trying to understand how much configurability, customization the buyer will tolerate?

[00:26:29] JN: Yeah. It's a good question. The answer I think largely depends on the maturity of the team and how big the team is as well. If you have one security person in the whole organization, more than likely they're just going to want to throw some product in that's fully configured and just is working. It's a little bit more complicated for a product like ours just because you really do have to tune it for your own business to really get the power out of it, right?

There're a lot of generalized standards that we can write [inaudible 00:27:00] rules for, and we've done that with Panther. I hired a security engineer last year and I made that his primary job, is just to research and write rules for detection and then policies for compliance. That's a really common ask people have, is like, "The customization is great and everything, but what do you give me out-of-the-box?" We say, "Oh! We give you about 150 rules and policies that cover CIS, they cover common S3 vulnerabilities. They cover EC2 vulnerabilities," like all of these really baseline checks that everyone should really be doing in any organization.

Then the beauty with Panther and StreamAlert is that you can just author your own. You can write your own detections on top. But the work that we've done at Panther in the last year or so is really helping those teams just going from zero to one saying like, "Hey, here is a pack of rules and policies. This will get you going, and then now you can start to build on top and you can tune them and customize them for your own needs."

[00:27:59] JM: Can you walk through the mental progression that you went through when you decided – You've been working on security tooling at Airbnb for more than 2-1/2 years, right?

[00:28:09] JN: About that.

[00:28:09] JM: Yeah. You were there for 2-1/2 years. When did you start to realize that the problems you were solving were generalizable and perhaps this is something that you should turn into a company?

[00:28:22] JN: Yeah. It's interesting. Because I started to think about my next steps in 2018. As I started to entertain options, I just found that kind of the same thing was happening in the other startups. Other hypergrowth companies, as you can imagine, there's a lot in the Bay Area started reaching out to me. I think as a security engineer, I think you're highly sought after if you're a security engineer. If you do detection and you can write code, you're highly valuable as an engineer, especially here in Silicon Valley.

I began to realize everyone has this problem and everybody knew that for a while. We have this bad tendency here in the valley to just build everything. I think that's most likely what I would have been doing if I had gone to another company. I think that idea was really like let's build something that has a company behind it. We have engineers dedicated full-time to it.

Because with StreamAlert, there's not people who are fully dedicated to that project. I mean, I don't know how it is today, because I've sort of disconnected from it. But when I was at Airbnb, I mean, I ran the team of engineers that was directly responsible for the project. Myself, I was that person for a longtime. My manager and I, we were the main people on the project, but I was the head engineer on it and I would work with other engineers in the company and we eventually grew our team, and then I stopped writing code and handed it to the team and just sort of managed the health of the team and everything in the higher level direction.

But the thing with Panther, it was like I wanted to start a company that was fully dedicated. So just working on this problem, and I guess we'll talk about in a second, like giving the project out for people to use it in all these other companies so they don't have to build it themselves. There is something reliable that they can just stand up in their AWS account that gives them all these benefits of being able to scale and do effective detection in a more mature platform in a mature way.

[00:30:21] JM: If I understand the chronology correctly, you'd built StreamAlert at Airbnb. It was open source. When you realized that this was a generalizable problem, you decided to start from scratch when you started Panther and build something similar to StreamAlert, but you just got to write everything from scratch. So it was a little bit of a cleaner implementation with all the knowledge that you had learned working on StreamAlert at Airbnb.

[00:30:53] JN: Yeah. That's essentially it. I mean, we really love the idea and the StreamAlert project itself was really successful and a lot of people in the industry really had a lot of use out of it. We knew that there's something there. When I started Panther, we wanted to really just start from scratch and kind of just rebuild from the ground-up and then rethink about the problem a little bit deeper.

I've talked with a lot of other security teams in the Bay since then and we've gone through a lot of iteration internally and tried to figure out what's the best product to really bring the market and to bring back into the community. That's what we're focused on this year. It's our going to market. It's kind of a good segue, but we're going to be open sourcing our platform soon. We're moving into a model that is – We have the core of Panther's open source. The essence is really the same. It's like a flexible modern cloud native SIEM is how I market it.

It's a platform that a team can standup in their account that will allow them to do real-time log analysis and allow them to do compliance as well, which is somewhat that's unique for Panther. StreamAlert doesn't really have any compliancy, sort of like first-class features built into it. It's more of a product of log analysis. But when we were designing Panther, we saw that that was such a common thing that we should build in like scanning resources in AWS accounts to make sure that they're compliant with X, Y, Z standards and we still use Python to do that evaluation. It's very, very similar. Then we also just have the traditional log analysis which is like we're looking for some activity. Let's detect that.

Yeah, we really want to start from scratch. Think about all of the use cases. Think about what an impactful product it would be and also learn from our experience. But the other thing about Panther that I love is that we've hired engineers from Amazon and Uber and even from Airbnb.

We've hired people who have like similar experience or maybe very different experience and have approached the problem in a very different way.

The two guys I hired from Amazon, for example, are bringing all of their knowledge into it and that has really shape how we built the platform and it changed a lot of the implementation actually that I had built at Airbnb. It's really fresh and new and I think it'll end up being even more scalable in the long-run than StreamAlert.

[00:33:16] JM: As you mentioned, this is a pretty non-standardized workflow that people are going to go through to get their logs analyzed, because in some cases, it's going to be logs coming off a database. In some cases it's maybe some system activity logs or logs, maybe they're taken out of Datadog or something like that. Does that ever happen, or is it actually standardized? Is it typically – Is there a common path that you can solve for in terms of where these log messages are getting aggregated or do you kind of have to do it customer by customer based on where they happen to be storing their logs?

[00:33:59] JN: Yeah, it's a good question. Luckily, there's a lot of standardization that happens within Amazon, and I would say our primary users – I mean, the primary users, the people that we target today are Amazon customers, like AWS users. Companies that have large amounts of infra within AWS.

There's a ton of standardized tooling there that we can natively plug into with cloud formation, which is just using infrastructure as code. It's very trivial to onboard data. In the ecosystem of AWS, it's easy. When you start to move on to like hybrid systems or databases, it becomes a little bit more complex, especially if you're outside of the scale of AWS. But it's till easy to do. You just have to have some type of logging pipeline in place. A lot of companies already have this. I mean, LogStash is very ubiquitous, in my opinion. I see that everywhere. Fluentd is also very common. Some people just use syslog itself and we can work with that as well.

Luckily, I actually come from a background of doing this at scale. To kind of like backup a little, when I was at Yahoo, I learned how to do DevOps at a very extreme scale, because Yahoo had an insanely big infrastructure. I had to learn how to deploy security tools to the entire infra and

then I had to learn how to aggregate all that log data back. Then that experience is what really helped me learn how to be successful in StreamAlert and then subsequently at Panther.

At the end of the day it's kind of a DevOps problem. Do you have instrumentation on all these places? If not, we'll guide you and help you set it up. At Panther, we've been writing a blog for the last 3 months, and we continually publish every week. Part of that is writing some of these topics that I'm talking about into a post an engineer can digest and say like, "Oh, okay. If I want to learn how to take my syslog data off of my VMs and AWS and put it in S3, this is how I could do it."

Actually, that's a real article that I wrote like a couple of weeks ago and it explains the process of using Fluentd and taking data from log files. All the configuration you need to make that work, and then we make it a series and we say, "Okay. Now to do a more complex version of this, you can have aggregation points and you can have failover," and like all these more like high-availability techniques that you can do.

A lot of it is kind of [inaudible 00:36:26] knowledge in my opinion, like setting up effective logging pipelines. It's not something that everyone's done before. Our whole effort with the blog is trying to democratize this data that's been like highly internal for so long, and we're also doing this for AWS logging too. I have a series going. I think we've done two so far. One was on CloudTrail, which logs all the API calls into your AWS account. Then another one was on S3 bucket logging, which is every type of request made to an S3 bucket, which as you can imagine is pretty impactful data to collect considering the amount of data breaches that have happened as a result of S3 misconfigurations.

[00:37:06] JM: Absolutely.

[00:37:08] JN: We're just trying to put all these info out in the open so we can empower security engineers to get a head start on all of these things, or maybe to teach them something new that they didn't know already. AWS has a ton of articles as well and they have a ton of information that's so helpful in their documentation, but sometimes it's so buried in there that you wouldn't really know where to look.

[00:37:29] JM: In terms of the market, do you feel like you're competing with any AWS logging solutions or with a Datadog or do you feel like this is its own product category?

[00:37:43] JN: I would closely relate it to SIEM as much as I don't like the connotation that the acronym generally has. But I think in order to –

[00:37:51] JM: What is the acronym again?

[00:37:53] JN: SIEM, security info and event management.

[00:37:57] JM: Okay.

[00:37:57] JN: Yeah. I think you could also compare – Like what I say in the beginning, most people use Splunk and Elasticsearch for this purpose. Generally, we're compared against that. But depending on where you focus in the product, so if you're looking at like compliance features, we would get compared against cloud security vendors, like RedLock, or even Evident, Evident.io, which was acquired by Palo Alto. It really depends on what the team is emphasizing, what they'll use the product for.

The primary purpose is for doing log analysis, in which case we're compared to traditional SIEMs, like God forbid, if people are still using like these really, really older SIEMs. But I would say, primarily, people would just build a SIEM on top of Splunk. That's usually like 9 times out of 10 what people are doing, or I guess now SumoLogic has a slightly easier platform than Splunk does. But in my mind, they're so similar.

[00:38:52] JM: Does that differ from your product, the Panther log monitoring product, because in the SumoLogic or Splunk world, they're hosting your logs also, right? You're shoving your logs into their systems. The Panther world is more decoupled because you're not actually storing them. You're just running analysis over their logs and they may be able to just keep their logs in on their own cloud servers. They don't necessarily need to be exported. Do I understand that product correctly?

[00:39:25] JN: Yeah, I'll explain it really quickly. I think with SumoLogic, they only have a SaaS implementation. You ship them the logs. It lives in one of their servers somewhere and then they give you a log-in where you can kind of go through the data. Splunk has a similar cloud offering, but they I think primarily do on-prem style deployment where the data stays within the premise of the company. That's the same model that Panther has.

With Panther, you would deploy within one of your AWS accounts. Generally, our recommendation is you create you a sub-account and then you deploy Panther within that. Then that's the account where you would send all your security data to. That's the account that you would run the Panther application itself, which will normalize the data and make sure it's in the correct state. It will store it in S3 and then it will do – We can use that data for compliance. We can use it for log analysis. We can also search over historically. That's really the area that we're going to invest more in as well, is like how can we understand the data better.

[00:40:27] JM: Take me through the full data pipeline from like some event that hits an application service, to the log message from that service being generated, to the log message being stored or just shuttled through Panther. I just want to understand the full data pipeline.

[00:40:47] JN: Sure. Yeah. From the source, let's just say that we're looking at security-focused applications. Meaning there are tools like OSQuery, which is operating system level instrumentation, and OSQuery will tell you which users do you have installed in your system, which programs do you have installed? What are some running processes? It just gives you any sort of generalized information.

With these tools, data can really be in any format. It could JSON data, it could be CS3 data, it could be some random custom format. In the case of syslog, for example, that could be literally whatever the administrator configures it to be. There can be a lot of challenges there.

But the first step in the pipeline is really just normalization, taking the data and putting it all in JSON. The reason we have to do that is because of the rules. They take in an event argument and then we write rules on it and it expects it to be in a certain format. We need to be normalized first as JSON, then we can run our rules on it and then we can generate alerts. But the way that pipeline works is the data is generated. It's aggregated. It's put into something

that's in the AWS ecosystem, whether it's a queue or an S3 bucket, or a Kenesis stream, and the recommendation we usually have is like S3 is probably the easiest way to get all these done and most reliable. So just use S3.

We pick it up from S3. We normalize it and then we send it through a series of pipelines, like Lambda functions that eventually have it land into a different S3 bucket in the right format. Then from there we do all our analysis on the data. We batch the data. We analyze let's say like a thousand lines at a time, right? Then any alerts that are generated sends into a different pipeline. Then maybe there's another fork of the pipeline that says, "Okay, let's use this for compliance. Let's scan any [inaudible 00:42:43] instance and make sure that all of these little attributes are the right setting. That's a common compliance workflow as well.

Then if there's an issue there, let's trigger another pipeline. The beauty with Lambda is that you can build these really complex architectures, but the gist is data comes in, it's normalized, it's analyzed and then alerts are send off. Then someone gets a text or a phone call on their phone. They click a link. It opens them into the Panther.ui. They can say, "Okay. This alert was generated. This is the full log file."
Then now we can go search our logs and find even more information." When we're searching logs, the backend for that is S3 and Athena and we're using these normalized logs that we set up to empower that pipeline.

[00:43:35] JM: Can you give me an example of how a customer has used it, like maybe a customer use case that exemplifies why Panther is useful.

[00:43:46] JN: Yeah. I would say onboarding any AWS level log data and then immediately getting alerts from it. Let's say – I was mentioning before, we have a bunch of built-in rules to the platform. Let's say they're already collecting their CloudTrail data, which is like a really common use case and everyone should be doing this. What they would do is they would – They'll install the Panther application in some account. They would link an S3 bucket to it and then we would start analyzing the S3 logs and then we could immediately tell you like, "Oh! This activity is happening right now," and it's really from this point in time onward.

Then alternatively, we can also use the same CloudTrail data to start to scan your infrastructure to say, “Okay. You actually have these five S3 buckets that you should look at right now as well. These are public buckets or they’re not following – They’re not encrypted or other configuration we would want to check,” and that’s kind of how the customer journey starts. It’s like we feed in one data source at a time. We see what alerts were generated. We tune the alerts and then we add more, and that’s the pattern we follow at Airbnb as well. We would start with some data. We would tune. We would write more rules. We would cover all these different use cases and then we’d move on to just onboarding more and more and more data. The more data we collect, the more well-rounded view that we have of everything.

The way I always explain it is like you could do cloud-based environment data, which is like your CloudTrail or your like infrastructure level logs, and then you move down the stack to like network logs. Layer three NetFlow data or if you have like layer 7, like full information on requests going into certain websites and their arguments and cookies and all that more rich information.

Then moving down the stack further, you have like host level information. It’s just stuff like OSQueries. It could be like databases. The things that are running the applications, which is the next level. Then applications could be like your web application. It could be like tools like OSQuery or Santa, which is a binary wireless thing tool that was developed by Google. You take all of these context and then you put it into a single place, and that’s how you sort of get the unified view of everything.

[SPONSOR MESSAGE]

[00:46:12] JM: Apache Cassandra is an open source distributed database that was first created to meet the scalability and availability needs of Facebook, Amazon and Google. In previous episodes of Software Engineering Daily we have covered Cassandra’s architecture and its benefits, and we’re happy to have Datastax, the largest contributed to the Cassandra project since day one as a sponsor of Software Engineering Daily.

Datastax provides Datastax enterprise, a powerful distribution of Cassandra created by the team that has contributed the most to Cassandra. Datastax enterprise enables teams to develop

faster, scale further, achieve operational simplicity, ensure enterprise security and run mixed workloads that work with the latest graph, search and analytics technology all running across hybrid and multi-cloud infrastructure.

More than 400 companies including Cisco, Capital One, and eBay run Datastax to modernize their database infrastructure, improve scalability and security, and deliver on projects such as customer analytics, IoT and e-commerce. To learn more about Apache Cassandra and Datastax's enterprise, go to datastax.com/sedaily. That's Datastax with an X, D-A-T-A-S-T-A-X, @datastax.com/sedaily.

Thank you to Datastax for being a sponsor of Software Engineering Daily. It's a great honor to have Datastax as a sponsor, and you can go to datastax.com/sedaily to learn more.

[INTERVIEW CONTINUED]

[00:47:53] JM: You're building tools around not just log management or log analysis, but also compliance. If I want to be compliant for SOC 2 or PCI, there are a number of things that I need to do to achieve that compliance. A lot of these has to do with configuration. Can you explain the connection between compliance and having the right resource configuration?

[00:48:26] JN: Essentially, for a resource to be compliant, it follows a lot of these checklists that the compliance standards outline. For example, there'd be a standard for encryption, is a really common one. Make sure that your data is encrypted at REST. What we can do with our product is we can scan a bucket. We can enumerate all of the attributes on it and then we can compare it against the policy. A policy defines desired secure state in a way.

Then we just develop a set of policies to cover all these different checks across all of these different standards, and then that's how we build the case for getting a compliance certification.

[00:49:04] JM: Those things are not built natively into AWS, like you can't say, "Hey, AWS, I want SOC 2 compliance across all my infrastructure."

[00:49:15] JN: No. You have to do some work to get there. AWS does have a tool called Config that will help you with this, but it's the same type of premise as with Panther, except with Panther you get more customization flexibility and it's more central than Config. With Panther, you can actually can like any number of accounts and it goes into one view, which is really helpful.

Then as a byproduct of that, you also get asset inventory too. You can kind of like go through all of your buckets and all of your accounts and you can say like, "Okay, are we failing any checks for CIS across any of our accounts, or do we have any buckets named X, Y, Z?" It's really helpful for that.

But out-of-the-box, yeah. I mean, AWS is getting better at this. They introduced a feature on S3 that will just block all public access. I'm not sure if that public access block aligns to like a compliance check per se, but yeah, they give some tooling, but it's still really on the engineer to do it.

[00:50:22] JM: That cloud asset indexing that you just described, there's a problem that I've seen in a couple of companies called Cloud Sprawl, basically, where they have all these resources that they don't even know where they are, where they existed. People throughout the organization have spun up databases and spun up servers and like these things are just not centralized in one place. There's no place you can go to see what are all my assets that users across my organization have spun up on AWS.

Can you describe that problem? Why is there a problem of indexing all of a company's cloud resources? Shouldn't there just be a place I go on AWS where I like click and like, "Here's everything I have."

[00:51:09] JN: Yeah. I mean, there should be, but there's not. I think it's because the accounts are all very segmented in a way. AWS is super powerful. You can do a ton within the confines of a single account. I think that's just an area they would need to invest in at some point and they probably will. I can't see it going on forever like this where you have this explosion of accounts happening and you need tooling around it. It hasn't really happened yet from AWS themselves. That just ended up being a byproduct of Panther.

We didn't really design the compliance product to say like, "Oh! We're going to build an asset inventory." It just kind of happened as a side effect. But it's a really good side effect, because now I can say like I can write a policy that says, "I only expect our users to ever exist in one of these accounts." Because you have that context across all the accounts, I think it provides you a better security control as well. You don't have to just like deploy your rules in a single account. You can look at all of them.

[00:52:11] JM: Okay. This is – Basically, the onboarding process is – Well, I guess in order to actually know whether you're compliant or not, you would have to wire the Panther compliance tool to all of your AWS accounts. Therefore, Panther is going to have a view into all those AWS accounts and know where all your resources are.

[00:52:33] JN: Yeah. The beauty with Panther is it's a single platform. We combine the compliance and log analysis into one. We actually use the same data for both in a way. Like I was saying before, the beauty of Lambda is you can create all these really fun complex pipelines, and that's exactly what we did.

For example, giving CloudTrail data, which is all the API calls that are happening in your AWS account, we can use that for log analysis. We can also use that to say, "Did some resource change?" We have a separate pipeline that gets – The data essentially forks into that says, "Okay. This log file shows that a user was created. Let's kickoff a scan on that user and get more information, and then let's compare the output of that into our policies, which map to compliance standards and then that can tell us if this user has too many permissions, for example."

Because another common compliance check is that you don't just have an admin attached to a user directly. You'd have it within a group or you wouldn't have it at all. I mean, having an administrator in AWS, like an administrator user is actually kind of seen as an anti-pattern, and you should use this privilege and that's a more common security best practice, right? That's really like the beauty with integrating.

Then when you integrate, you get the context across both too. If you get in a log event that happens that says like, “Jack did this thing.” Now I can actually see what permission does Jack actually have? What can you really do? Combine them into one platform with something else that was kind of new. You see it sometimes in SIEMs, but we really want to invest into that integration.

[00:54:09] JM: Let’s switch the conversation to focusing on the business aspect of this, the go-to-market. Are you trying to sell entirely to cloud native startups or do you feel like you can also sell to enterprises that have a mix of on-prem infrastructure and are getting started with the cloud or a year or two into their cloud product journey?

[00:54:34] JN: Yeah. I think it’s somewhat unrealistic if I was to say only cloud native companies, because I think any company of a significant scale will have both, will have on-prem systems and will have cloud-based systems. They might be multi-cloud. That’s a common thing too. Some people start dabbling in GCP because they like Bigtable or they can get the benefit out of using different clouds for different purposes.

[00:55:00] JM: Bigtable, GKE or TensorFlow.

[00:55:01] JN: Yeah, pretty much. Then with AWS, I mean, we try to focus on people who have large amounts of AWS infrastructure. But the beauty with logging tools is that they all natively support sending data to AWS anyway. Someone in a hybrid environment or someone in a purely AWS environment really wouldn’t make a difference to me just as long as they have some AWS infrastructure where they can run this application, where they can Panther itself. That’s really all that matters.

They can even have on-prem systems too, like laptops is a good example, right? Laptops really don’t have anything to do with the cloud. We can configure laptops to aggregate data into the cloud someway and then we can use the context outside of AWS. For example, like Palo Alto network firewalls or Cisco firewalls. Those don’t live in the cloud. I think each of those companies is working their way to bridge into the cloud. Yeah, I think the scenario of having on-prem appliances in data is so common that we have to support it.

[00:56:04] JM: The companies who historically have been on-prem and are cloud curious or they're getting into the cloud. Are they pretty adamant about wanting the security vendor to process the data on their infrastructure or are they open to sending the data to the security vendor?

[00:56:27] JN: I think 9 times out of 10, they wanted to stay within their premise, and there's a lot of reasons. I mean, data governance is one of them. Everyone's concerned with PII getting out and the more they can prevent that from happening, the easier they'll sleep at night kind of thing. It depends on a lot.

I mean, I think if you're a SOC 2 type 2 certified company or you're a very established business, it's easier to convince people to send you your data. If they're a really small team that maybe they literally don't even have the time to spin it up themselves, they would want something that's more SaaS-based. I think overall though, I really feel that most teams and companies don't want to send their data outside of the confines of their AWS accounts, which just makes it more comfortable.

[00:57:15] JM: What is the experience been like of developing a go-to-market strategy for a security company? It seems like selling a security product is this thing where it's really hard to get going, but then once you get going, once you have some case studies or proof points, it becomes much, much, much easier. What has been your experience for people listening who maybe are considering building a security company? What is the playbook look like?

[00:57:48] JN: The playbook is interesting. I think it really depends on who your buyer is or who your target audience is. Or us, it's engineers, security engineers. I mean, that's part of the reason why we wanted to go open core. We know that a lot of engineers are going to be deploying this and using this and customizing it. The more we can make them comfortable and empower them, I think the more successful we would be as a business, because we're building enterprise features on top of the open core model, which is a very common pattern you'll.

I mean, Elastic, for example, is like a classic one, right? They had an open source project, Elasticsearch. It was Apache 2.0 licensed. Then they ended up open sourcing all of their

enterprise features under their proprietary license. That was a highly successful company. They went public and they've done well. MongoDB is similar, right?

They had a slightly different approach with the same concept of let's open source our database platform and then let's encourage the massive adaption of it. I think that approach has been proven to work. We felt like there are a lot of other great benefits of that too, side effects of it, and that's you build a community and you're very transparent about what you're doing. In security, I think that's especially important to be transparent, especially when you're processing some company's extremely sensitive data.

You want to be transparent about like the way they were processing it, the fact that it all lives in your account and we just want to make people as comfortable as possible. We also follow just more of a scalable model to say, "Hey, Mr. Engineer, Ms. Engineer, who wants to deploy us? We don't need to be the blocker for you. You can go try it right now and just go download it from our website or go to our GitHub repo. Download it there. Just get going and use it without us."

I think if you're targeting engineers, that's probably the best place to start, is to have some flavor of open source or open core. If you want to build like a more traditional security platform, which a lot of companies do this as well, it all comes down to enterprise sales and just being able to kill it there and compete in the market.

[00:59:58] JM: You had been at Yahoo for 3-1/2 years before Airbnb. Yahoo was not a cloud native company. How did security at Yahoo compare to Airbnb?

[01:00:11] JN: There are a lot of similarities. I mean, the structure of the security org is pretty similar. You have people focus on application. You have people focused on instant response forensics. It was definitely a bigger team at Yahoo for obvious reasons. It was a bigger company. Been around for like 20 years. I think the mindset was the same though, because if you think about it, the security industry is so small. It's the same people running it everywhere. They all hop around. I'd say it's probably the same sort of mentality everywhere depending on if you're like at Facebook or at Stripe or some smaller growth stage company.

[01:00:47] JM: How is your experience starting your own company compared to working for a large employer?

[01:00:52] JN: You learn a lot, and as a founder, you have to – I don't know. You just have to know so many different things. You have to understand how to raise money, which is really difficult. You have to learn how to scale an organization. These are things like I still haven't really experienced first-hand yet. I mean, obviously, I raised a financing round last year and I – A lot of experience now there, and we could talk about that for a long-time. But the experience is night and day.

When I was a manager, I never had the same problems. Managing people is always hard. There's a lot of complexity in people and just building on those skills takes time and experience and mentorship and I think I've gotten better at that as well. But you just have to know everything. You have to know how to do anything as a founder, and that's really different, like marketing and going to market, and sales, and running a website, interacting with customers, pricing, fundraising, managing your board, managing the structure of the company, equity, all these things, payroll, benefits, health insurance.

You have to take care of your team. You have to make sure that they stay happy. You have to make sure you're building the right product. You have to look at the market and you have to understand what your competitor is doing. It's a really different shift, because as an engineer I'm just like, "I only care about the company I'm working for, really." With the case of StreamAlert, it was a little bit different because we opened sourced it and we worked a lot with other companies and we made a lot of friends that way. We had a lot of great relationships there and they contributed back to the project. We write customizations for certain companies and really do stuff that went into the open source to help them. But, yeah, it's definitely your mind expands quite a lot with the scope.

When you start talking with potential customers and users, you'd start to understand what their problems are. Instead of your own problems, you're now dealing with like everyone's problems and say like, "Okay. What are you struggling with? What is the most important thing for you right now?" They'll say, "Oh! Well, I have this problem and I need to get compliant, or I need to start analyzing our CloudTrail data or whatever."

You start to get in the minds of more teams, and I think that allows you to build a more well-rounded product when you have more context. It's kind of like when I hired the guys from Amazon, they're like, "Oh! We were doing it this way." I was like, "Oh! It's interesting. We're doing it this way at Airbnb." Then we converge on a better solution. The more people we can do that with, the more effective our product will be, and then thus like the more helpful it will be for everyone.

[01:03:26] JM: Last question. Are there any gaps in the security market that you think someone else should go out and work on?

[01:03:33] JN: I mean, besides the one that I'm doing, I don't' know. I think I have to think about that one a little bit deeper. I mean, the funny thing is, is like some of the biggest problems today have such simple solutions.

[01:03:47] JM: Like don't open that PDF.

[01:03:49] JN: Yeah. I mean, people are always the weakest link, right? I think there could be more work done in email. I think emails – I mean, funny enough, you mentioned that, but that's actually probably the most effective way.

[01:03:59] JM: That's the worst one, right?

[01:04:00] JN: It's the most effective way to compromise someone.

[01:04:02] JM: That's how the hospitals get ransomware. That's why your infosec team, even at Airbnb, is going to send you emails that say, "Hey, don't open the PDF."

[01:04:13] JN: Yeah. I think there's a company – I met this founder when I was in Israel. I went to Israel last November and I met a ton of founders and one of them runs a company that kind of takes a different approach to that problem where they want to secure the actual hardware themselves. I'm like blanking on the company name right now, but it's an Israeli cyber security company, and they want to address that problem by using like VMs that are kind of like air

gapped virtually like within – Actually, I think it's hardware air gapping. They do some really cool tech there to prevent compromise from happening.

You would have like a VM that's only for your email. You'd have a VM that's only for engineering and stuff like that, and then they can't really like communicate. That's like their effort to address that problem. But, yeah, in general, email security is another huge issue.

[01:05:01] JM: I think there's some really successful company – Well, there's at least one really successful company. I can't remember the name, but this idea of like you have to go through a VM to open or maybe even to do anything in the browser. I think they're like entire browserized, or basically like in order to use a browser at your enterprise, you open up the browser and then you basically log in to a VM and then the VM is your portal into a browser that's just running on that VM. Yeah, interesting.

[01:05:36] JN: Yeah.

[01:05:36] JM: Jack, thanks for coming on the show. Great talking to you.

[01:05:38] JN: Thanks for having me.

[END OF INTERVIEW]

[01:05:48] JM: Today's show is brought to you by Heroku, which has been my most frequently used cloud provider since I started as a software engineer. Heroku allows me to build and deploy my apps quickly without friction. Heroku's focus has always been on the developer experience, and working with data on the platform brings that same great experience. Heroku knows that you need fast access to data and insights so you can bring the most compelling and relevant apps to market.

Heroku's fully managed Postgres, Redis and Kafka data services help you get started faster and be more productive. Whether you're working with Postgres, or Apache Kafka, or Redis, and that means you can focus on building data-driven apps, not data infrastructure.

Visit softwareengineeringdaily.com/herokudata to learn about Heroku's managed data services. We build our own site, softwaredaily.com on Heroku, and as we scale, we will eventually need access to data services. I'm looking forward to taking advantage of Heroku's managed data services because I'm confident that they will be as easy to use as Heroku's core deployment and application management systems.

Visit softwareengineeringdaily.com/herokudata to find out more, and thanks to Heroku for being a sponsor of Software Engineering Daily.

[END]