**EPISODE 1318**

[INTRODUCTION]

**[00:00:00] JM:** According to Fugue's new state of cloud security 2020 report, cloud misconfiguration remains the top cause of data breaches in the cloud, and millions of databases are currently exposed across cloud providers. Some of the leading reasons are a lack of adequate oversight and too many API's and interfaces to govern. Argos security is a SaaS solution that detects and remedies misconfigurations of cloud assets within minutes. Argos detects and pinpoints exploitable issues in cloud environments, delivers a complete real time view of your cloud security posture. Gives you meaningful alerts with few false flags, and integrates in minutes with AWS, GCP and Azure.

In this episode, we talk with David O'Brian, founder of ARGOS Security.

[INTERVIEW]

**[00:00:49] JM:** David, welcome to show.

**[00:00:50] DO:** Thank you.

**[00:00:51] JM:** I'd like to start by getting your perspective on what modern software operations looks like. So we're in a time where you've got layer one cloud providers like AWS and GCP. You've got layer two cloud providers .You've got all kinds of Platform as a Service tools. What is your perspective on the modern state of software operations?

**[00:01:16]** DO**:** Yeah. Look, I think we are on this too speed world at the moment where you have the big brunt of people using cloud providers that are still only really getting started, even though it's over 10 years since the inception of clouds really. But then you have companies that are really just starting, or even parts of bigger companies that are starting to use the cloud as a more modern platform to build on. And that's where we've seen more uptake of Platform as a Services, Software as a Service, integration of Software as a Services. We see a lot of no-code happening at the moment where SaaS providers are being glued together through a platform.

 

And you see workflows happening and being created by people that aren't typically or traditionally been seen as developers even.

**[00:02:16] JM:** How has your perspective formed your product idea or informed what products you're building at ARGOS?

**[00:02:26] DO:** Yeah, that's really interesting, because I've been a consultant, IT consultant, cloud consultant for over 10 years. And I've always helped organizations get from where they were to a more modern stack, to a cloud stack, to modernize transform, buzzwords of digital transformation really. And you always fight for that big unicorn of everything fully automated, nobody touches production. Everything goes via pipeline, full DevOps, if you want to call it that. And at some point, I stepped back a bit out of that echo chamber that you quickly grow into of Twitter and social media in general, and stepped back and looked at the organizations that I've been working with for quite a long time and realized that not many people are actually at that point. A lot of people like I said, are still starting out with clouds, or trying to use the cloud now in a bit more cloudy way after they've just migrated or lifted and shifted, if you want, into the cloud.

And I realized that these organizations need a lot of support. And that support is very different to startup that's just starting out in the cloud and where they don't have any legacy software, they don't have any legacy applications that they need to take care of. There is no active directory or very timely, probably at this time, no printable as or print servers, or domain controllers that are being exploited. But most of our customers actually do have that. Most of our customers don't have cloud background. And that's really the organizations that at ARGOS we want to support by building a good cloud foundation.

**[00:04:34] JM:** What does that term mean, cloud foundation?

**[00:04:37] DO:** Yeah, it's really the boring stuff that a lot of organizations just skip over. It's not your fancy containerization, your fancy machine learning AI. It's probably not something that someone putting it into a job description is going to get a lot of applicants too, which is why a lot of people skip over it. It's the making sure that your cloud hygiene is being taken care of, that you don't expose things to the Internet. That you have your guardrails in place that makes sure that whoever is building something in your cloud environment is doing it in a secure way. And

secure means something different for each and every organization. Secure means something very different to the DoD probably compared to a small startup just trying to get their first customer. But we're trying to help especially enterprises with legacy applications trying to start out in the cloud, where they don't necessarily understand the shared responsibility that comes with public cloud that maybe have bought into some of the marketing of cloud providers that goes – That kind of implies, wrongly implies, that you use the cloud will take care of security, will take care of backups, will take care of disaster recovery and all of that, which is wrong. And then they don't do that. So these are the types of organizations that we want to head with their cloud foundations to actually build something that if the first wind hit, the first storm hits, it's not going to just topple over.

**[00:06:28] JM:** And what does that mean in practice, like on a technical level?

**[00:06:31] DO:** Yeah. So it means that your databases, for example, a lot of organizations go, and when they start in the cloud, they do try to modernize. So they're trying to not deploy virtual machines. So they're going to use something like Amazon RDS, Relational Database Services, or Database as a Service. But by default, these are all publicly exposed. They're all on the Internet by default. So what we make sure of is that that doesn't happen, that at least customers are aware of the security risk attached to having these services exposed to the Internet and have a quick way of fixing that and understanding why it's an issue.

It's the public S3 bucket that we read in every single data breach report almost that has been exposed and exposing data out to the public that people are actually aware of that issue. The firewalls that are instead of being misconfigured, and exposing RDP, which we've just learned, it's a massive attack vector for ransomware crews, for example. So all of these basic security hygiene that people in more modern environments almost take for granted in legacy environments that come from on-premises data centers where they don't have to worry about these things. That's what we head them with.

**[00:08:06] JM:** What's been your interaction with customers so far?

**[00:08:08] DO:** We have mostly enterprise customers, which kind of goes back to what I just said. They're the ones with legacy coming from data centers that do need that support of

building that good cloud foundation. They don't have the cloud background or the focus on cloud. And the interaction with our customers is very much an educational way of interacting with them. We have some white papers and blogs out that explains why it's actually an issue, that there is an issue and how to fix the issue. And that ARGOS is one way of fixing that issue.

But really, it's the education of we don't want to be seen as the product that goes and uses FUD, fear, uncertainty and doubt, to instill fear that people see cloud as the bad thing to use, because it's not. Arguably, in my opinion, you can build a much more secure environment in the clouds if you just follow these basic principles around security. And that seems to resonate quite well with our customers that we really want to help them instead of pointing fingers at them.

**[00:09:33] JM:** Say more about what that actually means.

**[00:09:36] DO:** I think a lot of people, especially when it comes to security, there's quite a lot of the cloud is insecure. But it's not. The cloud is not inherently insecure. It's whatever you make of it. And what we want to make sure is that people understand what they need to do in order to make it secure and that it's not – It requires a bit of change in the way they work to achieve that more secure cloud environment. But we do, for example, have had customers that went where the cloud isn't secure. So we're going to get back to on-prem. Because that's where it was more secure. That's where we had more control. And we're going to go back to on -premises. And that's where we want to educate our customers through our product and through trainings and interactions with them to understand the cloud is actually more secure. You have all the levers and buttons and everything you can do. You just need to do it. The cloud provider isn't going to necessarily do it for you. But you have to do it. And this is the way you do it.

**[00:10:52] JM:** Gotcha. So how does the sales process work for such a security company?

**[00:11:01] DO:** Yeah, so we unmask. Our sales process, it's very much self-service. People can go and sign up on our website for trial. There’s a free 30-day trial that they can sign up to. Usually they come on to our website after they've read a blog article or a whitepaper about cloud security. They come to the website, want to learn more, see what it's about, sign up to the trial, and then start using it. That's typically under 20 minutes to actually from sign up to getting data about your cloud security posture. That doesn't necessarily work with larger organizations.

Enterprises usually want a demo, and then a POC can sign up. But even there, it's usually a 20-minute, 30-minute demo, until they actually get data out of their environment. We typically do demos in their environment. So they just have to bring us some read only users so we can actually do the product demo live in their environment. And that's usually very convincing. They quickly see the issues in their environment and can right away respond to those issues.

**[00:12:21] JM:** And when you think about cloud security, and you're trying to design a product for cloud security, are you generally thinking of your target customer base as being on AWS? Or do you think more divergently? Or are you cloud agnostic?

**[00:12:38] DO:** We are cloud agnostic. So ARGOS works across Azure AWS and GCP, which is actually something that, as a cloud consultant in my previous life, always argued against. One cloud is difficult enough to manage. But we do see more and more customers of our ARGOS have more than one cloud provider, which is typically not necessarily for one application that they have multiple cloud providers, but one department uses AWS, another department uses Azure and so on. So yeah, we do see most of our customers on multiple clouds. And we can absolutely support that. And that's one of the features, I think, our customers really like that they do get the visibility across their whole cloud footprint in that one dashboard, which they typically wouldn't get through native tooling.

**[00:13:40] JM:** What are the problems with AWS from a security standpoint? Like if I want to focus on AWS for a second, what are the common security problems that people run into?

**[00:13:51] DO:** I think one of the issues is already that there's so many different security products on AWS. And it's not just AWS. Microsoft has the same thing, and their weird licensing around security, that security costs money. But it's the same on AWS. You don't get everything in one service in one tool. It'd difficult to consolidate all the information in one place just because of the way AWS has built their API's and the console. You can typically see things in one account across one region. And if you want to see something in a different region, you're losing the information from the other region. There are ways to consolidate, but these are then a bit more expensive again. And very quickly, you have to actually invest a lot of money into the native tooling that, yes, we'll help you with your security posture. But a lot of our customers seem to prefer by overbuild and by third-party products instead.

**[00:15:03] JM:** If you personally were building a web application today, would you build on AWS, or would you build on like a layer two cloud, or what exactly?

**[00:15:10] DO:** Good question. ARGOS itself it's hosted on Azure, not on AWS. So that is kind of an answer already. So we've chosen Azure. We will most likely go to AWS eventually as well for certain reasons that I can't really get into at this point. But I would probably for a web app use Azure again and not AWS. Azure is a lot friendlier, in my opinion, to web applications and to just getting it done than is AWS.

**[00:15:52] JM:** What's it like working in the Azure ecosystem? This is something I have not experienced firsthand. I've seen the AWS console. I've seen the GCP console. Actually, I have seen the Azure console one time, but it looked like it was kind of the same deal, just Microsoft-flavored, right?

**[00:16:08] DO:** No. I actually think if you need to use one of the three cloud providers, and you most of the time is going to use the console, you're probably going to be happiest on Microsoft, on Microsoft Azure. The information you get out of the dashboard is a lot more informative almost than it is on AWS. You get all the information about everything you have access to across all regions, across all subscriptions in one view, which is something you don't get on AWS. There's no way on AWS to see all your EC2 instances across all regions, across all the accounts you have access to in the EC2 console. On Azure, I can go into the console and say show me all the virtual machines across everywhere. And I can see all that information in one view.

I think from a management point of view, Microsoft is a lot friendlier to the human being than is AWS. Arguably, the CLI API experience on AWS is a bit more developer friendly, as it always has been. From a product vendor point of view, I still prefer Microsoft, because it's a lot easier to get information out of the Microsoft API standard as out of the AWS API's.

**[00:17:39] JM:** The suite of services on Azure. So in Google, you kind of have BigQuery, you have Cloud Run, you have Google Pub/Sub. In AWS, you have Fargate, Redshift, a bajillion

other things. Microsoft, you have CosmosDB, ACI. What are the other cool things you get from Azure? What are the other cool abstractions?

**[00:18:06] DO:** I do really like ACI. So Container Instances, which is the Fargate equivalent.

**[00:18:11] JM:** They beat Fargate to market. They were the first ones to have that product.

**[00:18:14] DO:** I think by few days. Yes.

**[00:18:16] JM:** No. I think was longer than that. It was around the same time. I think they had – When that product came out, I actually have – I gained a lot of faith in the Azure team.

**[00:18:26] DO:** Yeah, it is really good, because it's kind of what – Just like Fargate, you think here's a Docker image. I just want you to run this thing. Please do it. It's always what you think if you look at a container service, what you always thought you'd get, but you didn't ever get, because you had to worry about Kubernetes, and you had to worry about all these orchestrators. But all you really cared about was, can you please run this container for me? So I do like those services.

Apart from that, Microsoft – What I really like, one of their services is the Azure Functions service, which is like Lambda, only that it is a lot more geared towards people actually integrating into other services and that they don't have to worry about writing any integration code. As an example, if ARGOS wants to send an email and we want to send that email via SendGrid, then we don't have to write any code to write an email via SendGrid. It's just some metadata on that function that automatically integrates into SendGrid.

On lambda, we would have to write a whole codebase, boilerplate code, to tell Lambda how to send an email to SendGrid. So the Azure functions a lot nicer to people to actually write their own application. It's kind of what functions were always supposed to be, on Lambda, on Google Cloud, on Microsoft, that you just worry about your business logic. And that's all you have to worry about. And Microsoft does that very, very well. They had some issues, as always, with these services. But that's really my most favorite service on Microsoft Azure is the Azure Functions.

**[00:20:25] JM:** Is there anything about Azure Functions that's notably different than Google Cloud Functions or Lambda?

**[00:20:31] DO:** Yeah. Like I said, I think it's the input and output bindings, where they take care internally of integration into other things, which is what Lambda and Google Cloud Functions don't do. If you want to write a database entry from Lambda to RDS or on Dynamo, then you have to write that code to tell that lambda function which Dynamo database, how to authenticate. You have to use the SDK. Where to go? How to write it?

On Microsoft, you don't have to do any of that. It's just metadata you put onto that function. And that's it. It automatically knows how to authenticate to CosmosDB. How to Write to that database. You don't have to use the SDK at all.

**[00:21:25] JM:** Is that basically a better – Kind of a better API?

**[00:21:30] DO:** It's a lot nicer development experience in my opinion. And as someone creating a product, it means that we can actually focus on writing the product and not having to worry about integrating into things that, frankly speaking, don't make us money. Nobody's going to buy ARGOS because we know how to write emails via SendGrid very well, or how to write database entries into Dynamo very well. That's not a product feature. And so why do I need to worry about writing that code? Microsoft can take care of that for me. And that's really what I like about that.

**[00:22:15] JM:** As a security person, do you have any perspective on the various security properties of each of these clouds? Like is anyone notably better at security than the other ones?

**[00:22:26] DO:** I always say that you can achieve the same level of security on any of the three. And if you go and ask them or you go and look at their certification, they all have a bazillion certifications, compliance certifications. So you can achieve whatever you want on any of the three. The default of configuration, I think, it's probably the most secure on Google as I've seen. And Microsoft and Amazon are often insecure by default. And what I mean by that is if you go and look at Amazon, or Microsoft blogs, or tutorials, they want to get you started quickly. They

want to make sure that if you deploy an EC2 instance and install a web server, for example, on that EC2 instance, that you can get that started and done in minutes, not days, right? They want you to be able to connect to that EC2 to instance as quickly as possible, or that Microsoft Virtual Machine as quickly as possible, which means they're going to slap a public IP on to that. And they're going to export a firewall port out to the Internet so that you can go and connect to that.

Quite a few of these configurations that you go and look at in these articles actually allow all the ports to everybody, which means that in a matter of seconds, that resource that you just deployed is going to get smashed by the Internet and Internet scanners, and not just nice Internet scanners, that are going to try to fear that resource out and see if there's any holes in there that they can go and connect to. The getting started tutorials, however, are also quite often as I've seen in the past. The getting started and we're not ever going to look at this thing again once we've got started and deployed the thing that we needed to deploy for our business. So a lot of people don't necessarily go back to the thing they just deployed and secure it after the fact. So because cloud providers give you a bit less secure defaults to get started so you can get started quickly. That means there's a lot of insecure stuff out there that it's not going to get looked at again.

**[00:25:00] JM:** So most of the security companies I talked to are pretty well financed. Are you doing a bootstrap security company? Or are you doing venture financing? How are you thinking about the money aspect of the company?

**[00:25:14] DO:** Yeah. So we're fully bootstrapped at this point. We did get a bit of angel funding. So a few smaller checks. But at this point, we're fully bootstrapped. We are thinking about a potential seed round towards the end of the year or early next year, but that hasn't been officially announced yet. That is something that we're thinking about. At this point, bootstrapping the business is working well for us. And it's also keeping a bit of stress out of the company.

**[00:25:48] JM:** And what does it feel like to operate a bootstrap security company?

**[00:25:53] DO:** Well, I can tell you, I just bought a house. And that was very stressful, because banks don't understand what bootstrapping a company means. And so our savings and my co-founder savings have gone into this company. Bootstrapping, that means it's our money going

in. And it also means that there's a lot of skin in the game. It means there're a lot of emotions that go into this company. It, however, is also super rewarding. It's really, really great to get the feedback from customers how ARGOS has helped them increase their security posture. We've just had one feedback from a customer telling us that, “We just integrated ARGOS, and we've figured out, hey, there was some FTP server that somebody deployed months ago into a cloud environment that we didn't even know about that was exposing everything to the Internet.” One, it was costing them money. Two, it was a massive security issue. And that's super rewarding to hear that, that what you've paid, what I've paid, is actually helping people. And it's actually making things better. And if I was playing in quotes with somebody else's money, then sure, I would probably take a lot more risk with certain things. But it's kind of the rental car analogy. If it's not mine, then sure I can crash it into the wild and walk away, and the insurance is probably going to pay. But I just think it's very rewarding to be able to build this from the ground up over the last 12 months now, almost 18 months, and actually see it being successful.

**[00:27:47] JM:** That's really cool. I mean, I started a bootstrap company. I've really enjoyed doing it. I really have got to move at my own pace. I don't have anybody telling me what to do. We stayed at two employees for six years pretty much, me and one other person, some contractors. Yeah, we had a pretty good time. Bootstrap companies are pretty cool.

**[00:28:08] DO:** Exactly. And that's kind of our thinking at this point as well, that there is nobody telling us you have to do this, you have to go and set the rocket on fire, right? And absolutely explored the company, you have to fly first-class overseas, because you have to burn the money as fast as possible. Not that we're allowed to go overseas at this point, but least in Australia. But you're not having to justify every single spend that you're not doing. It's kind of as soon as it takes external money, it kind of flips. You have to justify why you're not spending money fast enough. Whereas at this point, we can make good decisions about what we want to spend money on. What does a good investment look like for us? And, yeah, we can go at our pace, and stay goal.

**[00:29:07] JM:** Under that regime, what is your strategic roadmap look like?

**[00:29:12] DO:** From a product point of view?

**[00:29:12] JM:** Well, it's like a product balance sheet point of view. Like how much resources do you have? What's your sales process look like? And given those constraints, how fast can you build new products?

**[00:29:22] DO:** Yeah, so we're very focused on that one product that we have, right? So it's the cloud security posture management, CSPM really. That's our core functionality, with a very, very big focus on security. We've just started development of that same capability for Infrastructure as Code. So that's going to be released soon, at least in the early version, so that we can scan and make the same judgment of your cloud security posture in your code, in your pipeline, in your CI/CD pipeline and not just for the running infrastructure, which I believe is more important to have that about your running infrastructure, and then your pipeline. But that's going to come next.

Our team at the moment is four people large. So we have four people working with ARGOS, that's two cofounders, so myself, a co-founder and two developers. Our sales process is very light touch, as I mentioned before. Most of our customers come to the website, sign up. Most of them watch the video.  Hopefully, when this airs, already updated website, and sign up to the trial, and start monitoring their cloud. All of that usually takes less than 20 minutes. And when or if somebody needs a demo, that's usually a 20 to 30-minute demo, we love doing those demos in the customer environment, if possible, so they can actually see in that demo, during that demo, everything like real data. It's not smoke and mirrors that you might sometimes see in vendor demos where somebody is actually just pressing play on a video, but it's a real thing. Otherwise, we just use the free environment and have our own demo environment in there. And yeah, like I said, it's very light touch. We don't have a sales team per se. We don't expect a thing to have a massive sales team, and be more around customer success after the fact.

**[00:31:36] JM:** Gotcha. You know what? I'd like to hear your perspective on what else is changing insecurity. For example, like a zero-trust networking I feel like is kind of a shift in ideology that's so deep and so pervasive that it requires a really broad suite of new products. I just kind of wonder if that's like a domain that you're at all interested in?

**[00:31:59] DO:** Yeah, personally, I'm definitely interested in all of that. The issue, I think, personally I have is that it's so broad that it includes our product. But there're so many other

moving parts to especially zero-trust that I just see a lot of our customers, a lot of organizations still struggle with the basic concepts of that. I mean, a lot of organizations are still using straight RDP or SSH access to virtual machines that are just using username and password or SSH keys to connect to virtual machines. They don't even have a VPN. There is no central authentication mechanism very rarely used.

We're seeing more and more interest in that. And more and more people are talking about it. But I do think it's going to be couple of years away until it actually gets mainstream and people actively working on implementing a zero-trust strategy into their environment. It's not impossible. We've seen Google do it, right? If a behemoth like Google can do it, then probably everybody can. But it really needs to be become a bit more mainstream. I do believe there need to be some more products that support this shift. And Google, I think just released – Or GCP just released a product, which is a set of principles that you can buy where they help you implement the zero-trust principles into your environment. But they also say that it is a big, not just technological shift, but also a big mindset shift, right?

One example, we're still getting a lot of security questionnaires from customers that ask us for our password rotation policy. Let's say, how complex is the password policy? How many characters do you enforce on your passwords? That just shows how far behind a lot of these organizations are when it comes to modern authentication, because we don't have really a username and password. We use password lists wherever possible to authenticate to services. We are already starting on the location-based and attribute-based authentication and authorization for ARGOS intended people. And I can tell you that's already difficult enough with certain contractors. And we're only four people now trying to implement something like that into an organization of thousands of people where a lot of them might not be technologically as mature as a software developer is. Yeah, there're a lot of issues I think that we as an industry is they have to overcome to actually make zero-trust a thing that's mainstream.

**[00:35:17] JM:** Give me your most breakthrough business revelation.

**[00:35:21] DO:** My most breakthrough business revelation is it doesn't matter what great ideas you as the founder have if customers don't care. So we had this idea, I had this idea for ARGOS. And the core idea hasn't changed at all. The core idea is still now what you see as a

product. It's still what the idea initially was. But there were certain things that we wanted to do that customers just didn't care about. And as a small business, as a startup, you really, really need to talk to your customers, talk to your early adopters before starting to build. And as a more technically-driven person, as someone who is much more hands-on, or used to be much more hands-on than talking to people, I'm probably that very stereotypical engineer, or developer, or sysadmin that really wants to start building something once they have that idea and then show it off to people once it's perfect. It really needs to be the other way around. It really has to be, when you have an idea, you talk to people about the idea. You take their feedback seriously. I think that's super important. And then you fit after that feedback. And then you start building once the feedback points towards, yep, there's interest for something.

It's super obvious, or it feels very obvious when you talk about it and once you've actually lift through that revelation. But before actually realizing that, I think it's a very – I think building a product is a very exposing thing in your life. Sounds a bit strange, but what I mean is you put something out there and you hope that people like it. It's like a book, or a painting, an artist, or a product and you hope people like it, and you don't want to, one, disappoint others, and you don't want to be disappointed. And that's why you spend weeks and months trying to perfect this thing before you put it out. Whereas, really, it should be the other way around, because otherwise you might be wasting months on something that people just don't want to buy. I think that's my biggest revelation.

**[00:38:04] JM:** Have you ever built products that nobody used?

**[00:38:07] DO:** Oh, there's plenty of GitHub repositories that nobody's ever used, I think. Plenty of ideas out there that I thought were the most amazing things and people go, "Not really." And, "Alright, here's some code, but nobody ever used it." Yeah, I think that happens all the time. Certainly, I don't have massive breakthrough ideas all the time. And I always thought of myself as somebody without a lot of ideas. And then ARGOS came around and popped up after something happened at a customer. And I thought, "Hmm, here's an idea. How we can do this better? And I started talking to people about this idea, whipped up an MVP, and put that in front of some people again. And they said, "Yeah, that's something." But on the same hand, I've had many, many, many people telling me about ideas I had, "No, that's a bit shit." Yeah, definitely.

**[00:39:14] JM:** Yeah, I've got like a lot of those. And it's great to build a graveyard of failed businesses, because then the graveyard always leads you to the goldmine.

**[00:39:25] DO:** Absolutely. And it's fun building something. It's always great to see or experience that beginning of something. It's sometimes disappointing if you really think this is something that people really, really need to care about, but people just don't. And then, sure, you can scream into the storm and nobody cares. But, yeah. No, I think if you have an idea, go build something after getting some validation from people, and at least put it out there on to GitHub or something similar and get some feedback. Get early feedback.

**[00:40:11] JM:** Well, that's seems like a great place to close off. David, thanks so much for coming on the show.

**[00:40:14] DO:** Thanks Jeffrey.

[END]